

Klinisch einsetzbare wissensverarbeitende Funktionen in einem onkologischen Informationssystem

Inaugural-Dissertation

zur Erlangung des Grades eines Doktors der Humanbiologie

des Fachbereiches Humanmedizin

der Justus-Liebig-Universität Giessen

vorgelegt von Ali Gerhard TAFAZZOLI

aus Calw / Baden-Württemberg

Giessen 1999

Aus dem Medizinischen Zentrum für Ökologie
Insitut für Medizinische Informatik

Leiter : Prof. Dr. J. Dudeck
der Justus-Liebig-Universität Giessen

Gutachter : Prof. Dr. J. Dudeck
Gutachter : Prof. Dr. von Lieven

Tag der Disputation : 21.10.1999

1. EINLEITUNG.....	5
1.1. HINTERGRUND	5
1.2. ZIELSETZUNG.....	6
2. DAS GIEßENER TUMORDOKUMENTATIONSSYSTEM (GTDS).....	7
2.1. KLINISCHE KREBSREGISTER UND DIE TUMORBASISDOKUMENTATION.....	7
2.2. DAS GIEßENER TUMORDOKUMENTATIONSSYSTEM (GTDS).....	9
3. WISSENSVERARBEITENDE FUNKTIONEN IN DER MEDIZIN.....	11
3.1. BEGRIFFSBESTIMMUNGEN	11
3.1.1. Wissensbasierte Funktionen	11
3.1.2. Wissensverarbeitende Funktionen	12
3.1.3. Entscheidungsunterstützung	13
3.1.4. Entscheidungsmonitoring	13
3.1.5. Wissensbasiertes System	14
3.1.6. Expertensysteme	16
3.2. VERTRETER WISSENSBASIERTER SYSTEME IN DER MEDIZIN	17
3.2.1. MYCIN.....	17
3.2.2. INTERNIST.....	18
3.2.3. Das HELP-System	18
3.2.4. Das CCC-System.....	19
3.2.5. Das Regenstrief Computing System.....	20
3.3. DIE KLINISCHE EINSETZBARKEIT WISSENSBASIERTER SYSTEME IN DER MEDIZIN.....	23
3.3.1. Gründe für die mangelnde klinische Einsetzbarkeit.....	23
3.4. DIE ARDEN-SYNTAX: EIN STANDARD ZUR WISSENSREPRÄSENTATION.....	27
3.4.1. Aufbau und Inhalte eines Medical Logic Modules	28
3.4.2. Eignung der Arden-Syntax zum Knowledge Sharing	31
3.4.3. Eignung der Arden-Syntax zum Knowledge Reusing.....	32
3.4.4. Beispiele für Arden-Syntax-basierte Systeme	33
4. SHELL FÜR WISSENSVERARBEITENDE FUNKTIONEN FÜR DAS GTDS	35
4.1. ANFORDERUNGEN AN DAS ZU REALISIERENDE WISSENSBASIERTE SYSTEM	35
4.2. AUSWAHL DER SOFTWARE	38
4.2.1. Oracle RDBMS Version 7	38
4.2.2. Oracle-Tools : Oracle SQL*Forms und PRO*C	41
4.2.3. Tools zur Compilerentwicklung : Lex und Yacc	41
4.3. NOTWENDIGE ERGÄNZUNGEN DER ARDEN-SYNTAX	43
4.3.1. Ergänzungen im Data-Slot.....	43
4.3.2. Modifikationen im Logic-Slot.....	47
4.4. ARCHITEKTUR DER SHELL FÜR WISSENSVERARBEITENDE FUNKTIONEN.....	49
4.4.1. Die Arden-Administration	49
4.4.2. Der Arden-Compiler	50
4.4.3. Die Arden-Engine.....	53
4.4.4. Der Arden-Melder	56
4.4.5. Technische Darstellung des Gesamtmoduls.....	58
4.5. UNTERSUCHUNG DER SHELL FÜR WISSENSVERARBEITENDE FUNKTIONEN	61
4.5.1. Die unterschiedlichen Wissenstypen innerhalb eines MLMs	61
4.5.2. Auswirkung der Wahl des Eventtyps auf den Logic-Slot.....	62
4.5.3. Auswirkung der Wahl des Eventtyps auf die Systemperformance	66
4.5.4. Knowledge Sharing versus Rückgriff auf separat abgelegtes Fachwissen	68
5. ANWENDUNG DER SHELL FÜR WISSENSVERARBEITENDE FUNKTIONEN.....	71
5.1. ANFORDERUNGEN AN ZU REALISIERENDE WISSENSVERARBEITENDE FUNKTIONEN	71
5.2. MÖGLICHE ANWENDUNGSGEBIETE DER ARDEN-SHELL INNERHALB DES GTDS.....	73
5.3. ANWENDUNGSFELD: SICHERUNG DER DATENQUALITÄT	75
5.3.1. Einleitung.....	75
5.3.2. Semantische Integritätsprüfungen als Methode zur Sicherstellung der Datenqualität.....	77
5.3.3. Realisierung von Integritätsprüfungen mit der Shell für wissensverarbeitende Funktionen ..	79
5.3.4. Umgebung zum nachträglichen Überarbeiten von Dokumenten	94

5.4. ANWENDUNGSFELD: STUDIENZUTEILUNG VON PATIENTEN	95
5.4.1. Konzeption der Unterstützung	95
5.4.2. Ein MLM zur Unterstützung einer Studienzuteilung	97
5.4.3. Zusammenfassung	100
6. KLINISCHER EINSATZ IM TUMORZENTRUM COTTBUS.....	101
6.1. MÖGLICHKEIT ZUM BEWERTEN UND NACHVOLLZIEHEN DER SYSTEMAKTIVITÄT	101
6.2. ERSTE ERGEBNISSE	103
6.2.1. Ergebnisse zum Meldungsaufkommen.....	103
6.2.2. Bewertung durch den Benutzer	106
6.2.3. Analyse einiger vom Benutzer als „inhaltlich falsch“ eingestufte Meldungen	108
6.2.4. Zusammenfassung der ersten Ergebnisse.....	111
6.3. BEWERTUNG DES ERSTEN EINSATZES	112
7. ERGEBNISSE	115
7.1. ERGÄNZUNGEN UND VERWENDUNGSHINWEISE ZUR ARDEN-SYNTAX	116
7.1.1. Ergänzungen am Sprachumfang der Arden-Syntax.....	116
7.1.2. Verwendungshinweise.....	116
7.2. ANWENDUNG DER ARDEN-SHELL IM GTDS	117
7.3. KLINISCHER EINSATZ IN EINEM TUMORZENTRUM	118
8. DISKUSSION	119
8.1. METHODISCHE ANFORDERUNGEN ZUR ERREICHUNG EINER KLINISCHEN EINSETZBARKEIT WISSENSVERARBEITENDER FUNKTIONEN	119
8.2. KLINISCHER EINSATZ	123
9. ZUSAMMENFASSUNG	125

ANHANG

A. GLOSSAR ZENTRALER BEGRIFFE	127
B. LITERATURVERZEICHNIS.....	128
C. REALISIERTE MEDICAL LOGIC MODULES	137
iarc0	137
iarc2	139
iarc3	142
iarc4	144
iarc5	146
iarc6	148
iarc10	150
iarc11	152
ip1	154
ip2	155
ip3	157
ip4	159
ip5	161
ip6	163
ip7	165
ST_IKNHL	168
ST_SMSGEM1	170
D. TABELLEN MIT DEM IARC-FACHWISSEN	172
E. SCHNITTSTELLE DES PROZEDURPAKETES ARDEN_PACK.....	173

meinen lieben Eltern ...

1. Einleitung

1.1. Hintergrund

Die Patientenversorgung wird durch die Tätigkeiten des im medizinischen Bereich kooperativ zusammenarbeitenden Personals gewährleistet. Ihre Arbeit wird durch die immer größer werdende Menge medizinischen Wissens und der damit einhergehenden Flut der bei einer Entscheidung zu berücksichtigenden Informationen erschwert. Dieser Sachverhalt wird durch die allgemeinen Schwächen des menschlichen Intellektes - nämlich das Übersehen von Begleitumständen, das Sich-Irren, die Abhängigkeit von psychologischen und physischen Faktoren [Goldberg 1970] - sowie den schwerwiegenden Folgen einer nicht optimalen Entscheidung noch verschärft. Als Ausweg aus diesem Dilemma werden seit Jahren Elektronische Informationssysteme im klinischen Umfeld eingesetzt. Zentrale Zielsetzungen dieser Systeme sind

- die Entlastung des medizinischen Personals von lästigen Routinearbeiten,
- die Erleichterung der Kommunikation zwischen den an der Behandlung beteiligten Personen und Einrichtungen sowie
- die Unterstützung der im Rahmen der ablaufenden Prozesse anfallenden Entscheidungsvorgänge durch das Zurverfügungstellen relevanter Informationen.

Abbildung 1 konkretisiert diese Zielsetzungen anhand des Gießener Tumordokumentationssystems (GTDS). Bei diesem System handelt es sich um ein Informationssystem für den Bereich der Onkologie, das mittlerweile an mehr als dreißig onkologischen Abteilungen bzw. Tumorzentren in Deutschland eingesetzt wird.

Entlastung	Kommunikation	Entscheidungsunterstützung
1. Arztbriefschreibung 2. Berichtgenerierung 3. Nachsorgeerinnerungen	1. Gemeinsames Krebsregister 2. Zentralregister Mal. Melanom 3. Zentrale Datenauswertung 4. Praxisdatensysteme 5. Klinikuminformationssysteme	1. des Arztes bei Diagnostik, Therapie, Studienzuteilung durch Abrufbarkeit von Patientendaten, Dosisberechnungen 2. des Dokumentars durch Wertelisten, TNM / ICPM-Browser

Abb. 1. : Konkretisierung der allgemeinen Ziele von Informationssystemen anhand des GTDS

Eine Unterstützung der Entscheidungsprozesse, die über das wunschgemäße Präsentieren von Informationen hinausgeht, erhofft man sich durch Systeme, die Kenntnisse über das betreffende Anwendungsgebiet besitzen. Hierdurch sollen diese in die Lage versetzt werden, das

medizinische Personal durch Hinweise auf Handlungsalternativen *während* oder durch Meldung von Entscheidungsfehlern *nach* dem Entscheidungsprozeß zu unterstützen. In dieser Arbeit sollen diese Systeme als wissensbasierte Systeme zur Entscheidungsunterstützung in der Medizin bzw. in Anlehnung an [Prokosch 1994] als wissensverarbeitende Funktionen bezeichnet werden.

Im Gegensatz zu den etablierten Informationssystemen haben die meisten dieser wissensbasierten Systeme hinsichtlich der Verwendung im klinischen Umfeld mit massiven Akzeptanzproblemen zu kämpfen. So stellt Wigertz [Wigertz 1986] fest, daß zwar einerseits immer mehr Systeme dieses Typs entwickelt wurden, aber andererseits überraschenderweise nur sehr wenige in der klinischen Routine eingesetzt werden. Eine Ausnahme hierzu stellt das an der LDS-Universität in Salt-Lake-City entwickelte HELP-System dar, in dem seit Jahren erfolgreich wissensverarbeitende Funktionen im klinischen Umfeld eingesetzt werden [Hripcsak 1995]. Die dort gesammelten Erfahrungen waren mitbestimmend beim Entwurf der Arden-Syntax. Hierbei handelt es sich um eine seit 1992 standardisierte (bei der „American Society for Testing and Materials“ als Dokument E 1460 innerhalb der Gruppe E31.15 Health Knowledge Representation [ASTM 1993]) Sprache zur Wissensrepräsentation für den medizinischen Bereich, von der man sich die Entstehung klinisch anwendbarer wissensverarbeitender Funktionen erhofft.

1.2. Zielsetzung

Der inhaltliche Schwerpunkt der Arbeit soll bei folgenden Punkten liegen :

1. die Synthese von Anforderungen an wissensverarbeitende Funktionen zur Erreichung einer Einsetzbarkeit im klinischen Umfeld
2. die Konzeption und Entwicklung eines auf der Arden-Syntax basierenden und an das GTDS gekoppelten wissensbasierten Systems
3. die schwerpunktmäßige Verwendung des Systems zur erfassungsnahen Unterstützung des dokumentierenden Personals bei der Entdeckung von Unplausibilitäten und Unvollständigkeiten
4. der Einsatz und eine erste Bewertung des Systems im klinischen Umfeld.

2. Das Gießener Tumordokumentationssystem (GTDS)

2.1. *Klinische Krebsregister und die Tumorbasisdokumentation*

Die Erkenntnis, daß die auf dem Gebiet der Krebsforschung und -bekämpfung arbeitenden Spezialisten auf den Erfahrungsaustausch und die Kooperation untereinander angewiesen sind, sowie die Notwendigkeit einer aufwendigen und langwierigen Versorgung von Krebspatienten, die eine ständige Überwachung, Nachsorge und Rehabilitation umfaßt, führte in den USA zur Errichtung von überregionalen Tumorzentren, deren Aufgabe unter anderem die Pflege von Tumorregistern ist. Nach diesem Vorbild wurden ab 1979 in der Bundesrepublik Deutschland sogenannte „Klinische Krebsregister“ eingerichtet. Während epidemiologische Krebsregister der Morbiditäts- und Mortalitätsstatistik dienen, sollen Klinische Krebsregister innerhalb eines Krankenhauses, Krankenhausverbandes oder einer ambulanten Praxis die Versorgung von Tumorpatienten verbessern helfen, indem die Zusammenarbeit zwischen den betroffenen medizinischen Fachbereichen bei der Krebsbehandlung erleichtert sowie die einheitliche Qualität bei der Versorgung von Tumorpatienten im gesamten Bundesgebiet gewährleistet wird. Derzeit werden Klinische Krebsregister von Tumorzentren an großen Krankenhäusern und Universitätskliniken sowie von Onkologischen Schwerpunkten an kleineren Krankenhäusern betrieben.

Die Klinischen Register verwalten und archivieren Patientendaten, Daten über Krankheitsverlauf, Therapie und Nachsorge, sowie Arztbriefe, Untersuchungstermine und anderes mehr. Um mit den gespeicherten Daten auch übergreifende wissenschaftliche Auswertungen und Analysen durchführen zu können, wurden 1979 die gespeicherten Datensätze durch einen Minimalsatz (Minimum Data Set), der „Tumorbasisdokumentation“ standardisiert.

Die Tumorbasisdokumentation verfolgt nach [Wagner 1983] vor allem folgende Ziele :

- Bereitstellung eines standardisierten, kodierten Datensatzes für jeden Tumorpatienten
- Vereinheitlichung der medizinischen Nomenklatur durch Anwendung international gebräuchlicher Schlüssel für Stadien und Diagnosen
- verbesserte Vergleichbarkeit von Daten und Befunden
- erleichterter Zugriff auf Patientendaten durch Kliniken
- schnelle und mühelose Erstellung von statistischen Übersichten

- Verbesserung der Kommunikation und Koordination zwischen Ärzten und Wissenschaftlern.

Da es sich zeigte, daß nach über zehn Jahren Einsatz der Tumorbasisdokumentation aufgrund bestimmter organisatorischer und merkmalsbezogener Probleme eine Überarbeitung notwendig ist, wurde 1990 eine Arbeitsgruppe gebildet, die die bisherige Tumorbasisdokumentation überarbeitet und neue Richtlinien festgelegt hat. Damit soll die Tumorbasisdokumentation den Anforderungen gerecht werden, die dadurch entstehen, daß die Basisdokumentation nun behandlungsnah erfaßt und gespeichert wird.

Die Daten der Tumorbasisdokumentation lassen sich nach folgenden Gesichtspunkten ordnen [Dudeck 1994]:

- Allgemeine Identifikationsdaten (inkl. Patientenstammdaten)
- Daten über die Erkrankung und den Zustand des Patienten
 - in der prätherapeutischen Phase (Diagnosedaten)
 - im Verlauf (Verlaufsdaten)
 - bei Abschluß der Betreuung (Abschlußdaten)
 - bei Autopsie nach Tod des Patienten (Autopsiedaten)
- Daten über vorgesehene und durchgeführte Maßnahmen (Therapiedaten)
- organisatorische Merkmale (Einbestelltermine etc.)

Da bei der Behandlung von onkologischen Patienten die Informationen aus verschiedenen medizinischen Bereichen (akute Labordaten, klinische Befunde, pathohistologische Befunde, Therapieparameter und deren Verlauf) schnell und vollständig verfügbar sein müssen, und weil das zu bewältigende Datenaufkommen oft beträchtlich ist, setzen praktisch alle Tumorzentren und Onkologische Schwerpunkte EDV zur Unterstützung der Tumordokumentation ein.

2.2. Das Gießener Tumordokumentationssystem (GTDS)

Im Hinblick auf den neuen Datensatz und den zu erwartenden Bedarf in den neuen Bundesländern wurde vom Bundesminister für Gesundheit die Entwicklung eines EDV-Systems gefördert, das auf der revidierten, neuen Basisdokumentation für Tumorkranke beruht und in Gießen unter Beteiligung zahlreicher Tumorzentren und Onkologischer Schwerpunkte entwickelt wurde. Das resultierende Gießener Tumordokumentationssystem (GTDS) stellt die notwendigen Funktionen sowohl für ein Klinisches Krebsregister als auch für eine Nachsorgeleitstelle zur Verfügung.

Folgende Funktion- und Leistungsbereiche werden vom GTDS unterstützt [Katz 1994]:

- Basisdaten/ Basisdokumentation
 - Stammdaten (Patient, Ärzte, Abteilungen)
 - Diagnosedaten
 - Verlaufsdaten
 - Abschlußdaten
- Therapiedaten
 - operative Therapie
 - Strahlentherapie
 - internistische Therapie/ Chemotherapie
- Arztbriefschreibung mit zahlreichen dynamisch erstellbaren Dokumenten
- Ausgedehnte Unterstützung bei der Nachsorge
 - Nachsorge mit festliegenden oder frei definierbaren Protokollen mit großer Flexibilität bei individueller Anpassung
 - Erzeugung der erforderlichen Dokumente (Einbestellbriefe, angepaßte Dokumentationsbögen)
- Zusammenfassende Auswertungen, einfache Statistik
- Datenaustausch mit dem Gemeinsamen Krebsregister in Berlin und organspezifischen Registern
- Systemumgebung und Werkzeuge

Das GTDS wurde aus Gründen der technischen Portabilität im wesentlichen mit Oracle - Werkzeugen und der Programmiersprache C realisiert. Es existieren derzeit circa 30

Installationen in Unix - Umgebungen auf den gängigen Hardwareplattformen. Neben der ursprünglichen zeichenorientierten Version ist seit einigen Monaten eine graphische Version im Einsatz. Entwicklungsschwerpunkte liegen - abgesehen von der Erweiterung um wissensverarbeitende Funktionen - in einer Komplettierung des Funktionsumfangs der graphischen Version, in der Realisierung von Datenaustauschnittstellen über HL7 bzw. BDT und in der Entwicklung einer internetbasierten Version.

3. Wissensverarbeitende Funktionen in der Medizin

Bereits in Kapitel 1.1. wurden wissensverarbeitende Funktionen als Funktionen charakterisiert, die aufgrund von Kenntnissen über ein spezielles Anwendungsgebiet in der Lage sind, das medizinische Personal durch Aufzeigen von Handlungsalternativen *während* oder durch Hinweise auf Entscheidungsfehler *nach* dem Entscheidungsprozeß zu unterstützen. Gegenstand dieses Kapitels ist zunächst eine Präzisierung sowie die Abgrenzung zu verwandten Begriffen. Danach werden einige Beispiele wissensverarbeitende Funktionen realisierende Systeme vorgestellt. Anschließend werden die Gründe für das häufige Scheitern dieses Typs von Systemen im klinischen Umfeld analysiert. Das Kapitel endet mit einer kurzen Einführung in die Arden-Syntax, von der man sich die Entstehung klinisch anwendbarer wissensverarbeitender Funktionen in der Medizin erhofft.

3.1. Begriffsbestimmungen

3.1.1. Wissensbasierte Funktionen

In vielen Arbeiten werden die Begriffe „wissensbasierte Funktionen“ und „wissensverarbeitende Funktionen“ als Synonym verwendet. Betrachtet man die zwei Begriffe allerdings etwas genauer, so fällt auf, daß das Adjektiv „wissensbasiert“ intuitiv eher einen passiven Charakter hat, wohingegen „wissensverarbeitend“ mehr auf eine direkte Aktion hindeutet. Dies berücksichtigend, soll im folgenden „wissensbasiert“ auch eher den deklarativen Charakter des reinen Beschreibens von Wissen und „wissensverarbeitend“ den mehr prozeduralen, mit einer aktiven Anwendung des Wissens verbundenen Charakter einer Funktion bezeichnen. In Anlehnung an [Prokosch 1994] können wissensbasierte Funktionen folgendermaßen definiert werden.

„Wissensbasierte Funktionen sind diejenigen Mehrwert erzeugenden Funktionen, die den Übergang von Daten auf Informationen ermöglichen. Hierbei kann es sich zum einen komfortable Transformations-, Retrieval-, und Aufbereitungsfunktionen handeln, die es ermöglichen, aus einer Vielzahl der innerhalb eines Krankenhausinformationssystems gespeicherten Daten mittels Extraktion, Kombination und Aggregation die zur Lösung eines gegebenen klinischen oder administrativen Problems relevanten Informationen zu gewinnen und dem Benutzer aufbereitet darzustellen. Zum anderen handelt es sich um Funktionen, die es ermöglichen, eher

globales Fakten, wie z.B. allgemein anerkanntes medizinisches Fachwissen an den Datenverarbeitungs-Arbeitsplätzen eines Krankenhauses online verfügbar zu machen.“

Als typische Beispiele für wissensbasierte Funktionen können also „klassische“ Präsentationsfunktionen von Krankenhausinformationssystemen und medizinische Wissensserver innerhalb des Internets genannt werden. Allen diesen Funktionen ist gemeinsam, daß sie die menschlichen Entscheidungsprozesse *im Vorfeld* der Entscheidung durch Präsentation relevanter Daten unterstützen.

3.1.2. Wissensverarbeitende Funktionen

„Wissensverarbeitende Funktionen“ sind in der Lage, durch Anwendung ihnen bekannten Wissens eigenständig Entscheidungen abzuleiten. Sie können nach [Prokosch 1994] folgendermaßen präzisiert werden.

*„In einer **wissensverarbeitenden Funktion** wird explizit repräsentiertes Wissen aktiv auf aktuell gegebene Fakten angewendet. Hierbei kann entweder der Prozeß der menschlichen Entscheidungsfindung unterstützt („Entscheidungsunterstützung“) oder aber eine getroffene Entscheidung mit vorliegendem Wissen verglichen und dadurch abgesichert werden („Entscheidungsmonitoring“).“*

Der Wert dieser Definition liegt in der umfassenden Charakterisierung wissensverarbeitender Funktionen. Zum einen beschreibt sie auf einem relativ anwendungsnahen Niveau das Verhalten wissensverarbeitender Funktionen (Entscheidungsunterstützung bzw. Entscheidungsmonitoring), zum anderen legt sie auch eine wichtige methodische Eigenschaft fest, nämlich, daß eine solche Funktion auf explizit repräsentiertem Wissen aufsetzen muß. Die Eigenschaften expliziten Wissens werden im Zuge der Begriffsbestimmung „wissensbasiertes System“ dargestellt. Aufgrund dieser letzten Eigenschaft ist das eine wissensverarbeitende Funktion realisierende System immer auch ein wissensbasiertes System.

3.1.3. Entscheidungsunterstützung

Zur vollständigen Beschreibung wissensverarbeitender Funktionen gehören die mit dieser Definition verknüpften Definitionen für Entscheidungsunterstützung und Entscheidungsmonitoring [Prokosch 1994].

„Entscheidungsunterstützende Funktionen setzen in einem frühen Stadium des menschlichen Entscheidungsprozesses an. Der Mensch muß dem System in der Regel Vorinformationen über seine Entscheidungssituation liefern und wird dann durch einen Entscheidungsvorschlag vom System bei der Festlegung seines Handlungsplans unterstützt. Diagnosesysteme sind typische Beispiele entscheidungsunterstützender Funktionen.“

Besonderes Augenmerk soll der Formulierung „bei der Festlegung seines Handlungsplans unterstützt“ gelten. Diese Unterstützung variiert bei existierenden Systemen von einer symbiotischen Zusammenarbeit von EDV-System und Benutzer bis hin zur Degradierung des Benutzers zum passiven Teilnehmer an einer Systemsitzung. Diese Spannweite der Rolle des Benutzers bzw. EDV-Systems bei Entscheidungsunterstützenden Funktionen ist Grundlage der in Kapitel 3.3.1. vorgenommenen Einteilung existierender System in die Klasse der monolithischen und katalytischen Systeme.

3.1.4. Entscheidungsmonitoring

*„Mit dem Begriff **Entscheidungsmonitoring** bezeichnet man Funktionen, die vom Anwender bereits getroffene Entscheidungen im Hintergrund überwachen. Solange die vom Anwender getroffenen Entscheidungen durch das im System definierte Expertenwissen abgesichert ist, nimmt er die Ausführung der Monitoringfunktionen nicht wahr. Diese werden lediglich dann nach außen hin aktiv, wenn sie einen Verstoß gegen das gespeicherte Expertenwissen feststellen. In einem solchen Fall können sie den Anwender über das vorliegende Problem informieren und somit diesen dazu anregen, die getroffene Entscheidung noch einmal zu überdenken.“*

Kritisch zu dieser Definition ist die Beschränkung auf Expertenwissen anzumerken. Viele sinnvolle auf relativ trivialem Allgemeinwissen basierende wissensverarbeitende Funktionen

sind denkbar, die die allgemeinen Schwächen des menschlichen Intellektes - das Übersehen, das Sich-Irren, die Abhängigkeit von psychologischen und physischen Faktoren - kompensieren. Ledley bemerkt hierzu treffend „most errors are errors of omission“ [Ledley 1959].

3.1.5. Wissensbasiertes System

Der Begriff wissensbasiertes System ist auf den ersten Blick irreführend, da er dazu verleitet anzunehmen, daß als solche benannte Systeme, sich durch das Beinhalten von Wissen von konventionellen Systemen (Systeme, deren Verhalten vollständig durch prozeduralen Code beschrieben ist oder - anders formuliert - deren Wissen in impliziter Form repräsentiert ist) unterscheiden. Dies ist jedoch nicht der Fall: selbst in konventionellen Systemen ist Wissen enthalten, weil Wissen eine notwendige Voraussetzung für die Durchführung eines Ablaufes oder Algorithmus darstellt. Auch die Annahme, daß prinzipiell nur mit einem wissensbasierten System komplexere Probleme, wie beispielsweise das Diagnose- oder Therapieproblem, gelöst werden können, ist falsch. Es ist zwar sehr aufwendig, aber durchaus möglich, die Problemlösung eines wissensbasierten Systems durch ein konventionelles Programm zu realisieren. So verweist beispielsweise Szolovits darauf, daß zur Beschreibung des komplexen Verhaltens von MYCIN - einem frühen wissensbasierten System zur Diagnostik und Therapie schwerer bakterieller Infektionen - ein konventioneller Algorithmus, der zwar einige hundert Seiten Flußdiagramm umfassen würde, denkbar wäre [Szolovits 1982].

Das Charakteristische an wissensbasierten Systemen ist also nicht, daß sie Wissen enthalten, sondern *die Art*, wie dieses Wissen im System enthalten ist. Konventionelle Systeme enthalten Wissen in impliziter, wissensbasierte Systeme in expliziter Form. Friedrich und Stumpner formulieren in [Gottlob 1990] den Unterschied zwischen implizitem und explizitem Wissen folgendermaßen:

„Je nachdem, ob dieses Wissen im Programmcode versteckt und verteilt ist, oder ob es in bestimmten Teilen des Systems lokalisierbar ist, spricht man von implizitem oder explizitem Wissen“. ... „Implizites Wissen steckt z.B. im Code eines Programmes, das eine bestimmte mathematische Funktion berechnet, z.B. das Produkt zweier Matrizen. Der entsprechende Algorithmus wird nicht aufgerufen, weil das Programm nach dem Algorithmus gesucht hat, sondern weil der Programmierer wußte, daß er ihn im

Programmablauf an dieser Stelle anwenden muß. Das Wissen, daß dieser Algorithmus an einer bestimmten Stelle (d.h. bei Erreichen einer bestimmten Anweisung) ausgeführt werden muß, ist also im Programmcode versteckt. Ein Prozeduraufruf ändert daran nichts. Das Wissen ist zwar abstrahiert im Unterprogramm versteckt, aber die Information, welches Unterprogramm aufgerufen werden soll, ist wieder fix im Programmcode enthalten.“ ... „Im expliziten Fall sind dem System seine Problemlösungsmöglichkeiten für eine bestimmte Aufgabe bekannt, also an einer bestimmten Stelle im System explizit vermerkt und werden den Bedürfnissen entsprechend angewendet.“ ... „Systeme, die Wissen in expliziter Form enthalten, werden als wissensbasierte Systeme bezeichnet.“

Das für konventionelle Systeme typische implizite Wissen liegt verteilt und unlokalisierbar vor. Seine Existenz kann nur indirekt durch das Verhalten des Systemes festgestellt werden. Bei solchen Systemen erfordern kleine Änderungen des Wissens oft weitreichende Eingriffe in das gesamte Programmsystem, die oft unerwünschte Effekte haben können. Deshalb ist die Wartung solcher Systeme in der Regel schwierig, aufwendig und fehleranfällig.

Im Gegensatz dazu sind im Fall des Vorhandenseins expliziten Wissens, im System die Problemlösungsmöglichkeiten für eine bestimmte Aufgabe an einer bestimmten Stelle explizit vermerkt.

Weiterhin kann zwischen deklarativem und prozeduralem expliziten Wissen unterschieden werden :

- deklaratives explizites Wissen : es wird beschrieben, was getan werden muß, d.h. in welchem Verhältnis Einheiten der betrachteten Welt zueinanderstehen. Weiterhin existiert eine isolierte aktive Komponente - eine sogenannte Inferenzmaschine - , die dieses Wissen zur Durchführung der eigentlichen Problemlösung anwendet.
- prozedurales explizites Wissen : es wird in unmittelbar ablauffähiger Form angegeben, wie eine Aufgabe gelöst werden soll. Der Unterschied zwischen explizitem und implizitem prozeduralem Wissen liegt darin, daß die Beschreibung der Auswahl eines bestimmten Algorithmus' nicht prozedural sondern deklarativ erfolgt.

Die Vorteile einer expliziten Wissensdarstellung liegen letztlich in einer erleichterten Wartung und Erweiterbarkeit des vorhandenen Wissens. Durch diese Flexibilität wird eine Anpassung an den aktuellen Stand der Wissenschaft oder an neue Erfordernisse praktisch erst möglich.

3.1.6. Expertensysteme

Im Zusammenhang mit wissensbasierten Systemen fällt oft der Begriff Expertensystem. Als Expertensystem wird ein Computerprogramm bezeichnet, das in einem eng abgegrenzten Gebiet mit dem Wissen eines (menschlichen) Experten ausgestattet ist und auf diesem Gebiet dessen Problemlösungsfähigkeit erreicht oder übertrifft [Haux 1987]. Expertensysteme werden meist wissensbasiert konstruiert. Während der Begriff Expertensystem den externen Aspekt des Systemverhaltens beinhaltet, betont der Begriff wissensbasiertes System den internen Aspekt der Systemstruktur. Das eigentlich Besondere an einem Expertensystem in Abgrenzung zu wissensbasierten Systemen liegt in der hohen Komplexität des inkorporierten Wissens. Dieses Wissen wird meist in explizit deklarativer Form dargestellt. Daher werden sie oft auch als eine Teilmenge der wissensbasierten Systeme verstanden. Anstelle des Begriffes „Expertensystem“ wird immer häufiger der Begriff „wissensbasiertes System“ verwendet, da ersterer zu übertriebenen Einschätzungen über die Funktionsweise des Systems verleitet.

In anderen Begriffsbestimmungen zu wissensbasierten Systemen ist eine starke Orientierung am Begriff des Expertensystems deutlich. Dies äußert sich typischerweise in einer Gleichsetzung von prozeduralem Wissen mit implizitem Wissen. Eine Differenzierung zwischen implizitem und explizitem prozeduralen Wissen wird im Gegensatz zu der in dieser Arbeit verwendeten Begriffsbestimmung nicht vorgenommen.

3.2. Vertreter wissensbasierter Systeme in der Medizin

Seit Anfang der siebziger Jahre herrscht ein reges Interesse an der Entwicklung wissensbasierter Systeme in der Medizin. Auslöser hierfür war vor allem der Wunsch, ein anspruchsvolles Anwendungsgebiet für den zuvor entstandenen Forschungszweig der Künstlichen Intelligenz zu finden. In diesem Kapitel soll exemplarisch an ausgewählten Systemen die Vielseitigkeit dieser Systeme hinsichtlich Anwendungsbereich, Methodik, Problemen bei der klinischen Einsetzbarkeit und der dem Anwender zukommenden Rolle gezeigt werden.

3.2.1. MYCIN

Das an der Stanford University entwickelte *MYCIN* [Shortliffe 1976] ist das bekannteste wissensbasierte System im medizinischen Bereich und Vorbild zahlreicher anderer Systeme. Sein Anwendungsbereich liegt in der Diagnostik und Therapie schwerer bakterieller Infektionen. Aus der Analyse bereits gespeicherter Informationen über einen Patienten entwickelt MYCIN einen Dialog mit dem Anwender, in dem die zur Diagnosestellung noch fehlenden Daten erfragt werden. Die komplexe Fragestellung konnte mit dem MYCIN - Paradigma auf nur einige hundert verständliche Regeln und einen einfachen Rekursionsalgorithmus reduziert werden. In MYCIN wird das Fachwissen in Regeln dargestellt, und die Diagnostik wird durch zielgerichtete, rückwärtsverkettete Auswertung der Produktionsregeln durchgeführt. Die in dem Fachwissen und den Evidenzen beinhalteten Unsicherheiten werden in MYCIN mit Sicherheitsfaktoren („Certainty Factors“) statisch beschrieben und dynamisch behandelt. So ist zu jeder Produktionsregel eine heuristische Verknüpfungsfunktion zum Aggregieren der Sicherheitsfaktoren der Evidenzen vorhanden. Diese Funktion stellt eine Approximation des klassischen Bayes-Theorems dar. Es gilt so für die Verknüpfungsregel die Voraussetzung unabhängiger Evidenzen. In den häufigen Fällen, in denen diese Voraussetzung nicht erfüllt ist, versucht man resultierende Anomalien durch geschicktes Anpassen der Verknüpfungsfunktionen und Sicherheitsfaktoren zu vermeiden („Tuning“) [Gottlob 1990 b]. Aus MYCIN wurde durch Entnahme des Fachwissens die erste Expertensystemschale EMYCIN [Jackson 1987] und durch Hinzufügen von anderen spezifischen Funktionen das Tutoring-System GUIDON [Clancey 1984]. Die Verwendung von MYCIN als Expertensystemschale zeigt die allgemeine Anwendbarkeit von Produktionsregeln. Formale Bewertungen von MYCIN haben gezeigt, daß es im Bereich der Diagnostik und

Therapie von Bacterämie und Meningitis dem Entscheidungsverhalten von menschlichen Experten sehr nahe kommt [Yu 1979 a], [Yu 1979 b]. Mycin wird jedoch nicht in der klinischen Routine verwendet, da seine Wissensbasis bezüglich der Diagnostik und Therapie anderer Infektionskrankheiten unvollständig ist. MYCINs besondere Rolle besteht darin, daß es das erste ernstzunehmende entscheidungsunterstützende System im medizinischen Bereich darstellt.

3.2.2. INTERNIST

INTERNIST [Miller 1982] ist ein Expertensystem für den Anwendungsbereich der inneren Medizin. Es wurde von Pople und Myers an der Universität Pittsburgh entwickelt. Es besteht im wesentlichen aus zwei Komponenten: einer Wissensbasis und einem heuristikbasierten diagnostischen Algorithmus. Die Beziehungen zwischen Krankheiten und Symptomen werden in Form von Krankheitsprofilen in der Wissensbasis repräsentiert. Ein solches Profil ist eine Liste von allen bei einer gegebenen Krankheit auftretenden Symptomen, wobei jedem Symptom zwei numerische Faktoren, die sogenannte „Evoking Strength“ und „Frequency“, zugeordnet sind. Die Wissensbasis von INTERNIST-I bildet auch noch pathophysiologische, nosologische und andere mitwirkende Relationen ab. In der Wissensbasis von INTERNIST-I waren schon bald ungefähr 600 Krankheitsprofile und über 4000 Symptome bekannt, womit eine Abdeckung von 75 % der Diagnosen der inneren Medizin erreicht werden sollte. Auf der Basis klinischer Symptome generiert das System Differentialdiagnosen. Hierbei spielen Punktezubeweisungen sowie Punkteabrechnungen („SCORING“) eine Schlüsselrolle. Komplexere Problemformulierungen, wie bspw. die Diagnose eines Patienten, der an mehreren Krankheiten leidet, führten zu Problemen und zur Entstehung weiterer Versionen. Nachfolger von INTERNIST-I sind INTERNIST-II, CADUCEUS und QMR („Quick Medical Reference,“) [Miller 1986]. Die Bedeutung von INTERNIST für die Entwicklung von wissensbasierten Systemen liegt darin, daß aufgezeigt wurde, daß bei vielen Problemen ein reines Oberflächenwissen, wie es bei INTERNIST-I vorliegt, nicht ausreicht, sondern für eine Problemlösung ein tieferes Verständnis für die inneren Zusammenhänge nötig ist [Gottlob 1990 c].

3.2.3. Das HELP-System

Das *HELP-System* [Pryor 1983] ist kein wissensbasiertes System im Sinne der bisher vorgestellten Systeme. Es ist vielmehr ein klinisches Informationssystem, das das medizinische

Personal bei Entscheidungen unterstützen kann. Die Hauptmerkmale von HELP sind die Erweiterungsfähigkeit der zugrundeliegenden Patientendatenbank um neue Merkmale, die Fähigkeit, Logik zur Unterstützung von Aufgaben aus dem medizinischen und administrativen Bereich zu verarbeiten, sowie die Beinhaltung von Subsystemen für Fragestellungen aus dem Bereich der Forschung. Das Wissen zur Entscheidungsunterstützung ist im HELP-System modular aufgebaut. Es kann als WENN-DANN-Regeln oder als probabilistische Regeln abgelegt werden, die auf dem Bayes-Theorem basieren. Die Anwendung des Wissens auf konkrete Problemsituationen erfolgt „data-driven“, d.h. daß letztendlich die Speicherung von Daten die Wissensmodule aktiviert. Dabei kann die wissensbasierte Komponente benötigte, aber nicht vorhandene Daten über das Erfassungsmodul des HELP-Systems vom Benutzer erfragen. Die Ergebnisse der Wissensverarbeitung können zu verschiedenen Zielen, wie bspw. die Patientendatenbank, Stationsdrucker oder spezielle Forschungsdateien, geschickt werden.

3.2.4. Das CCC-System

Safran stellt in [Safran 1996 a] das sogenannte *CCC-System* als eine Erweiterung des Informationssystems des „Center for Clinical-Computing“ am Bostoner Beth Israel Krankenhaus vor. Zielsetzung des Systems ist die Unterstützung der Behandlung HIV-infizierter Patienten. Typisch für diesen Bereich ist neben der Komplexität der Behandlung, die große Dynamik im Wissen zu Diagnostik und Therapie [American College of Physicians 1994]. Weiterhin müssen im Falle einer Unterstützung, eventuelle Informationen zeitnah zum Patientenkontakt bereit stehen. Vorarbeiten zur Systementwicklung waren unter anderem eine Befragung von mit der Behandlung von HIV-Infizierten erfahrenen Ärzten. Die vorgeschlagenen Leitlinien beinhalten die Bereiche

1. der Sammlung von grundlegenden Daten zum Krankheitsmonitoring,
2. der Einleitung präventiver Maßnahmen,
3. des periodischen Screenings und
4. des Therapiemonitorings.

Wesentliche Bestandteile des Systems sind sogenannte „Alerts“ und „Reminder“. Hierbei handelt es sich um für den Anwender bestimmte Meldungen. Der Unterschied zwischen „Alerts“ und „Reminder“ liegt hierbei lediglich in der Art und Weise, wie der behandelnde Arzt informiert wird. Informationen, die als besonders wichtig eingestuft sind, werden als „Alert“ unmittelbar nach ihrer Generierung präsentiert. Als „Reminder“ eingestufte Meldungen werden unter Umständen erst nach einer Zeitverzögerung dem behandelnden Arzt übermittelt. Die

Informierung erfolgt typischerweise, wenn der Arzt Bestandteile des entsprechenden Patientendatensatzes konsultiert bzw. bearbeitet. Ein besonderes Merkmal des Systems ist neben der unmittelbaren Präsentation hilfreicher Informationen die Unterstützung der Durchführung einer eventuell geforderten Aktion. Wenn die Meldung dem behandelnden Arzt vorschlägt, die Dosis eines speziellen Medikamentes zu modifizieren, so bietet das System die Möglichkeit,

- die Dosis zu modifizieren (mit der Möglichkeit, einen neuen Behandlungstermin zu vereinbaren, ein Rezept auszustellen oder einen Patientenbrief zu schreiben),
- die Meldung als unpassend zu kennzeichnen,
- die Meldung einer anderen Person zu übermitteln,
- die Anzeige der Meldung auf einen anderen Termin zu verlegen oder
- Patientendaten anzuzeigen.

Zur Bewertung des Erfolges wurde eine achtzehnmonatige Studie durchgeführt, in der untersucht wurde, ob das Verhalten der Ärzte, die keine Meldungen erhielten, von dem der Ärzte, die Meldungen erhielten, hinsichtlich der Erfüllung der definierten Leitlinien differiert. Die veröffentlichten Ergebnisse [Safran 1996 b] belegen, daß im Bereich der Behandlung von HIV-Infizierten die Präsentation von „Alerts“ und „Reminders“ zu einer signifikant schnelleren und vollständigeren Anpassung an praktische Handlungsleitlinien führten. So wurde beispielsweise festgestellt, daß bei 338 Alerts, die in der Kontrollgruppe generiert worden wären, die durchschnittliche Zeit bis zum Zeitpunkt der geforderten Aktion bei 52 Tagen lag, während sie in der mit Meldungen versorgten Gruppe lediglich elf Tage betrug.

Das vorgestellte System bietet weiterhin die Möglichkeit, auf unterschiedliche Online-Informationsbanken zuzugreifen.

3.2.5. Das Regenstrief Computing System

Bei dem ab 1972 schrittweise entworfenen *Regenstrief Computing System* [Mc Donald 1992] handelt es sich um ein Informationssystem, das mittlerweile Daten von mehr als 500.000 Patienten in unterschiedlichen Einrichtungen der Stadt und Umgebung Indianapolis beinhaltet. Als besondere Eigenheit soll hier auf die integrierten sogenannten „Reminder“ eingegangen werden. Hierbei handelt es sich um kleine Programme, die aus zwei Teilen bestehen. Einem ersten Teil, der eine oder mehrere durch die booleschen Operatoren AND und OR Bedingungen enthält, und einem zweiten Teil, der aus einer Nachricht besteht, die bei Erfüllung der Bedingungen des ersten Teils ausgegeben wird. Die Programme, die letztlich Regeln

darstellen, sind in der CARE-Language geschrieben. Die Sprache CARE beinhaltet arithmetische Funktionen, die Möglichkeit einen Wert in Abhängigkeit zu einem klinischen Ereignis (z.B. the first creatine kinase level after the last heart attack) zu bestimmen und die Möglichkeit einzelne Regeln zu einem komplexeren Gebilde zu kombinieren. Die auf diese Weise definierten „Reminder“ kontrollieren typischerweise vor einer angesetzten Visite die vorhandenen Patientendaten, in der Absicht den Arzt an Sachverhalte, die ärztlicher Kontrolle bedürfen könnten, zu erinnern. Die im RMRS - System verwendete Namensgebung „Reminder“ ist prägend geworden für Meldungen, die erst nach ihrer eigentlichen Entstehung aufgrund eines äußeren Ereignisses (wie ein Betrachten der Daten zum entsprechenden Patienten, oder vor der Durchführung der Visite) präsentiert werden. Momentan werden zirka 600 solcher Regeln verwendet. Einige exemplarische Anwendungsbeispiele für diese Reminder sind,

- auf einen unbehandelten Hypothyreoidismus oder Anormalitäten hinzuweisen,
- in bestimmten Fällen Mammographien im Rahmen einer Vorsorgeuntersuchung anzuregen,
- die Bestimmung des Idealgewichts oder des Framingham Cardiovascular Risk Indexes und
- Präventivmaßnahmen zu veranlassen [Mc Donald 1976 a], [Mc Donald 1976 b].

Für die seit 1974 eingesetzten „Reminder“ wurde eine randomisierte kontrollierte Studie an 12.000 Patienten durchgeführt [Mc Donald 1984], die einen statistisch signifikanten Effekt der Reminder auf die Patientenbehandlung nachweist.

3.3. Die klinische Einsetzbarkeit wissensbasierter Systeme in der Medizin

Der Nutzen von computerunterstützter Entscheidungsfindung wurde das erste Mal in den späten fünfziger Jahren prognostiziert [Ledley 1959]. Nach mehr als dreißig Jahren Forschung und vielen tausend wissenschaftlicher Veröffentlichungen haben jedoch nur sehr wenige Systeme das Prototypstadium verlassen. Heathfield führt eine europaweite Untersuchung an, nach der lediglich sechs Systeme eine routinemäßige Nutzung erlangt haben [Heathfield 1993]. Ein vergleichbarer Sachverhalt besteht in den Vereinigten Staaten sowie in Japan [Goodyear 1990].

3.3.1. Gründe für die mangelnde klinische Einsetzbarkeit

In der Literatur sind viele Gründe für die noch nicht ausreichende Akzeptanz genannt worden.

- Ein häufig festgestellter Grund ist die unzureichende Integration in klinische Arbeitsabläufe und vorhandene Informationssysteme [Spitzer 1994], [Mann 1995], [Miller 1990]. Miller verweist hier exemplarisch auf die zwischen 30 und 90 Minuten dauernden und damit für die klinische Routine inakzeptablen Konsultationssitzungen des Systems Internist-1.
- Typische Probleme der Wissensbasis sind ihre Vollständigkeit und Konsistenz. Die Erfahrungen zum System MYCIN haben gezeigt, wie schwer es ist einen vollständigen Wissensbestand zu erreichen. Feigenbaum spricht in diesem Zusammenhang von der Wissensakquisition als Flaschenhals bei der Entwicklung wissensbasierter Systeme [Feigenbaum 1977]. Bezüglich der Konsistenz stellt Alvey exemplarisch an einem System zur Unterstützung der Leukämiebehandlung fest, daß es keineswegs trivial ist, eine Menge von Regeln über ein bestimmten medizinischen Teilgebiet zu definieren, so daß keine unerwünschten Nebeneffekte auftreten [Alvey 1987].
- Weiterhin sind oft - bedingt durch die hohe Komplexität der verwendeten Schlußfolgerungsmechanismen - die Systemfunktionsweise für den Anwender nicht nachvollziehbar und die gebotenen Erklärungsversuche für den Benutzer unplausibel [Spitzer 1994].

Die folgenden Gründe zeichnen sich dadurch aus, daß es sich bei ihnen um Grundsatzentscheidungen handelt, die bereits in einem sehr frühen Stadium der Entwicklung wissensverarbeitender Funktionen zu einer Weichenstellung hinsichtlich der klinischen Einsetzbarkeit führen. Die weiter oben genannten Gründe dürften sich in der Regel auf diese zurückführen lassen.

- **Motivation der Entwickler:** Bei der Entwicklung vieler Systeme stand das Ausprobieren einer neuen KI-Idee im Vordergrund, wobei der Praktikabilität des klinischen Einsatzes unberücksichtigt blieb [Shortliffe 1993]. Eine solche Motivation kann selbstverständlich durchaus seine Berechtigung im Rahmen von grundlagenorientierten Forschungsprojekten haben, kann jedoch im strengeren Sinne nicht als Forschung im Bereich der Medizinischen Informatik bezeichnet werden [Haux 1990]. Es kann jedoch nicht ausgeschlossen werden, daß mitunter der Aufwand der Einführung eines Systems beim Anwender oder konkret in der klinischen Routine unterschätzt wurde. In diesem Zusammenhang soll auf die bewährte 90:10 Daumenregel des Software Engineering verwiesen werden, nach der zehn Prozent des Aufwandes während der Lebensdauer eines Systems in der Entwicklungsphase und neunzig Prozent nach Einführung beim Anwender im Rahmen von Wartung und Anpassungen an neue Gegebenheiten verbraucht werden.
- **Wahl des Anwendungsgebietes:** Leider werden oft Anwendungsgebiete ausgewählt, in denen von vornherein keine Akzeptanz zu erwarten ist. Hierfür können zwei unterschiedliche Gründe verantwortlich sein.
 1. Es werden Anwendungsgebiete gewählt, in denen niemand unterstützt werden will. Nach Avron besteht oft in den gewählten Anwendungsgebieten wissensbasierter Systeme keine Notwendigkeit diese auch einzuführen, da die Ärzte ohne sie mindestens genau so gut zurechtkommen [Avron 1982]. Heathfield schlägt wieder den Bogen zur Motivation der Entwickler, indem er behauptet, daß diese oft die Tätigkeitsbereiche der möglichen Endbenutzer unberücksichtigt lassen und so Bereiche sinnvoller Entscheidungsunterstützung übersehen. Als Folge stimmen die von den Systementwicklern gewählten Anwendungsbereiche nicht mit denen überein, die Kliniker als problematisch empfinden [Heathfield 1993]. Er begründet diese Behauptung mit einer MEDLINE-Untersuchung, nach der das Hauptinteresse bei

Nachfragen im therapeutischen Bereich lag, während sich Systementwickler im wesentlichen auf den diagnostischen Bereich konzentriert hatten.

2. Es werden Anwendungsgebiete ausgewählt, in denen vorhandene Methoden unzulänglich sind. Blois stellt in diesem Zusammenhang fest, daß die für das medizinische Personal problematischsten kognitiven Prozesse oft auch am besten für eine Unterstützung durch wissensbasierte Systeme geeignet, jedoch nicht unbedingt interessant bezüglich einer wissenschaftlichen Untersuchung sind [Blois 1980]. Leider wird nicht konkretisiert, welches diese kognitiven Prozesse sind oder wieso hier der Einsatz wissensbasierter Systeme als besonders geeignet erscheint. Andere Autoren haben einige Charakteristika medizinischer Entscheidungsfindung herausgearbeitet, die eine maschinelle Nachbildung extrem erschweren, wenn nicht sogar unmöglich machen. So weist Groß darauf hin, daß beim diagnostischen Problemlösen neben dem in Form von Lehrbüchern und Literatur zugänglichem medizinischen Wissen Intuition, Erfahrung und die Einsicht über die Grenzen des eigenen Wissens eine Schlüsselrolle bei der Entscheidungsfindung spielen [Groß 1969]. Anfängliche Versuche, diese Aspekte durch Metaregeln abzubilden sind gescheitert [Davis 1984]. [Anschütz 1987] verweist auf die hohe Inter- und Intravariabilität medizinischer Entscheidung, und wirft damit die Frage auf, welche Problemlösung ein wissensbasiertes System - sollte es existieren - vorschlagen sollte. Weiterhin weist er darauf hin, daß beim diagnostischen Entscheidungsprozeß aufgrund potentieller Multimorbidität sowohl krankheitsübergreifendes Wissen als auch aufgrund der Tatsache, daß eine Diagnose sogleich Therapieindikation ist, tätigkeitsübergreifendes Wissen notwendig ist. Weitere Charakteristika menschlicher Entscheidungsfindung sind die Berücksichtigung unbewußter oder auch schwer operationalisierbarer Fakten. Miller und Masarie verweisen hier darauf, daß ein wissensbasiertes System ggf. mit einem im Vergleich zum behandelnden Arzt nur sehr bruchstückhaften Bild über den Patienten auskommen muß [Miller 1990]. Basierend auf dieser Einsicht fordern die Autoren, daß wissensbasierte Systeme lediglich mit einer „katalytischen“ Zielsetzung unter Berücksichtigung des Wissens des Benutzers konzipiert werden sollten. Dieser Aspekt ist Gegenstand der folgenden Grundsatzentscheidung.

- Wahl der Rolle, die dem Anwender zukommt. [Miller 1990] differenziert wissensbasierte Systeme in monolithische und katalytische Systeme. Monolithische Systeme tendieren hierbei dazu, dem Arzt die Entscheidungsfindung abzunehmen, während katalytische Systeme lediglich beabsichtigen, dem Arzt durch Hinweise (Leitlinien, Handlungsalternativen, Erkennen typischer Fehler, Alarmsituationen) bei der Entscheidungsfindung zur Hand zu gehen. Katalytische Systeme führen zu einer positiven Akzeptanz im klinischen Umfeld. Als Beleg soll auf die Systeme der Kapitel 3.2.3. bis 3.3.5. verwiesen werden, die der katalytischen Kategorie angehören und klinisch akzeptiert sind. Abbildung 2 stellt die typischen Merkmale dieser Systeme gegenüber.

Monolithische Systeme	Katalytische Systeme
Der Entscheidungsprozeß läuft innerhalb des Systems ab, der Benutzer wird weitgehend zum passiven Teilnehmer an einer Sitzung.	Der Entscheidungsprozeß liegt beim Benutzer. Er wird durch das System an bestimmten Stellen dieses Prozesses durch Hinweise (Richtlinien, Leitlinien, Handlungsalternativen, Erkennen typischer Fehler, Alarmsituationen) angeregt. („Katalysierung des menschlichen Entscheidungsprozesses“). Der Benutzer kann <ul style="list-style-type: none"> • entsprechend seinem individuellen Stil, • unter Verwendung seines Fachwissens, • seiner zum Teil unbewußten oder schwer operationalisierbaren Kenntnissen zum konkreten Fall • und seiner persönlichen Erfahrung und Intuition den Entscheidungsprozeß durchführen.
Das System tritt mit dem Selbstverständnis an das gesamte relevante Wissen in vollständiger, konsistenter Form zu besitzen. Wissen des Benutzers wird ignoriert.	Das System enthält typischerweise Wissen über typische Fehler des Benutzers und über eng umrissene Bereiche.
Das System funktioniert im Stile einer Black-Box. Durch die schwere Nachvollziehbarkeit seiner Funktionsweise kann der Endanwender nur sehr bedingt Einfluß auf die Systemweiterentwicklung nehmen.	Die geringere Komplexität der Funktionsweise erleichtert es dem Endanwender, Einfluß auf das Systemverhalten zu nehmen. Im Idealfall stellt das System einen Werkzeugkasten zur Entscheidungsunterstützung für den Endanwender dar.
Das System muß versuchen durch Erklärungen den Benutzer von der Korrektheit seines Entscheidungsprozesses zu überzeugen.	Eine Erklärungskomponente wird nicht benötigt. Im Normalfall führt das Systemverhalten zu „Aha-Effekten“ beim Benutzer.

Abb. 2: Gegenüberstellende Charakterisierung monolithischer und katalytischer Systeme

3.4. Die Arden-Syntax: Ein Standard zur Wissensrepräsentation

Die Arden-Syntax stellt eine Wissensrepräsentationssyntax dar, die im Rahmen eines 1989 in Arden Homestead (Columbia University, New York, USA) durchgeführten Workshops formuliert wurde. Sie wurde mit der Intention entwickelt, einmal dargestelltes Wissen mit möglichst geringem Aufwand zwischen unterschiedlichen Einrichtungen auszutauschen und so die Wissensakquisition als Flaschenhals bei der Entwicklung wissensbasierter Systeme zu überwinden. Seit 1993 wird die Arden-Syntax von der ASTM als Standard geführt [ASTM 1993].

Die Arden-Syntax erlaubt den Aufbau von Wissensbanken, die aus voneinander unabhängigen Wissensmodulen bestehen. Mit dieser Strukturierung orientiert sich die Arden-Syntax am Repräsentationsformalismus klinisch etablierter Systeme, wie des HELP Systems oder des Regenstrief Computing Systems. Ein unabhängiges Wissensmodul enthält hierbei das *gesamte* nötige Wissen, um innerhalb eines Informationssystems *eine* konkrete wissensverarbeitende Funktion im Stile von Entscheidungsunterstützung oder Entscheidungsmonitoring durchführen zu können. Diese Strukturierung ist eine Folge der Forderung nach möglichst problemloser Austauschbarkeit formulierten Wissens. Mit ihr geht jedoch auch eine leichte Verifikation und Wartung der wissensverarbeitenden Funktionen einher. Beispiele konkreter wissensverarbeitenden Funktionen sind

- die Sicherstellung der Dokumentationskonvention, nach der es zu einem Tumorfall nur eine Haupthistologie geben darf,
- der Hinweis auf mögliche Kontraindikationen bei einer speziellen Medikation,
- der Hinweis auf eine nötige Dosisreduktionen bei einer speziellen Medikation aufgrund von übersehenen Nierenerkrankungen oder
- die Sicherstellung der Integritätsbedingung, daß bestimmte Histologien mit einigen Lokalisationen unverträglich sind.

Wissensbasierte Systeme, die die Arden-Syntax zur Wissensrepräsentation verwenden, sind typischerweise eng an ein vorhandenes Informationssystem gekoppelt. Diese Kopplung äußert sich darin, daß resultierende wissensverarbeitende Funktionen automatisch nach Auftreten eines speziellen Ereignisses im zugrundeliegenden Informationssystem ausgelöst werden

(Ablaufintegriertheit) und auf schon im Informationssystem enthaltene Daten zurückgreifen (Datenintegriertheit). Die Konfrontation mit Hinweistexten ist für den Benutzer das einzige Indiz für die Existenz der wissensverarbeitenden Funktionen. Die enge Kopplung des wissensbasierten Systems an ein Informationssystem wird durch die Beinhaltung von auslösenden Ereignissen und Angaben über relevante Fakten des Informationssystems in einem Wissensmodul möglich.

Die hohe Integration in ein vorhandenes Informationssystem, die Eignung zum Wissensaustausch zwischen unterschiedlichen Institutionen und die leichte Verifikation und Wartbarkeit einzelner Wissensmodule lassen die Arden-Syntax als vielversprechenden Ansatz zur Erreichung klinisch einsetzbarer, wissensverarbeitender Funktionen erscheinen.

Die Arden-Syntax selber ist eine in Backus-Naur-Form - einer häufigen Beschreibungsform kontextfreier Grammatiken - vorliegende Sprachbeschreibung zur Darstellung der zueinander unabhängigen Wissensmodulen. Diese in der Arden-Syntax-Terminologie als „Medical Logic Modul“ (im folgenden als MLM abgekürzt) bezeichnete Wissensmodule lassen sich in Kategorien zerlegen, die wiederum in Unterbestandteile - sogenannten Slots - weiter aufgeteilt werden können. Abhängig vom jeweiligen Slot kann der Slotkörper Freitext oder andere durch die Arden-Syntax festgelegte Konstruktionen enthalten. Die Arden-Syntax stellt letztendlich eine Kombination deklarativer und prozeduraler Wissensrepräsentationsmethoden dar.

3.4.1. Aufbau und Inhalte eines Medical Logic Modules

Ein MLM besteht aus maximal neunzehn Slots, die aus Gründen der besseren Lesbarkeit in die drei Kategorien „Maintenance Category“, „Library Category“ und „Knowledge Category“ unterteilt sind. Während die ersten beiden Kategorien mehr dem Wissensbankmanagement und der MLM-Dokumentation dienen, findet man das eigentliche medizinische Wissen in der „Knowledge Category“. Innerhalb dieser findet sich auch der prozedurale Anteil der Arden-Syntax, da medizinische Entscheidungsregeln in der Form von Wenn-Dann-Verknüpfungen dargestellt werden können. Abbildung 3 gibt eine Übersicht über die in einem MLM enthaltenen Kategorien und Slots.

3.4.1.1. Die Maintenance und Library Category

Die erste Kategorie wird als *Maintenance Category* bezeichnet. Sie enthält Verwaltungs-
informationen, wie beispielsweise den Titel oder den Urheber zur Identifikation der
Herkunftsquelle des MLMs. Dies ist insbesondere bei Austausch eines MLMs zwischen
unterschiedlichen Einrichtungen wichtig. Im Validation-Slot können die Werte „testing“,
„research“, „production“ und „expired“ zur Beschreibung des Verwendungsstatus verwendet
werden. Wenn ein MLM ausgetauscht wird, sollte dieser Slot auf „testing“ gesetzt werden, um
darauf hinzuweisen, daß die empfangende Institution den Einsatz unabhängig von der
sendenden Institution untersuchen sollte. Das gleiche gilt für den Specialist-Slot, in dem der
Verantwortliche für das MLM innerhalb der einsetzenden Institution eingetragen werden sollte.

Die *Library Category* enthält Angaben über den Zweck und eventuell zugrundeliegende
Publikationen. Um innerhalb einer größeren Wissensbank aus MLMs gezielt auf relevante
MLMs zugreifen zu können, ist hier innerhalb des Keywords-Slot die Angabe von
Schlüsselwörtern möglich. Im Explanation-Slot kann eine freitextliche Erläuterung über den
Zweck und die Arbeitsweise erfolgen. Diese kann beispielsweise als einfache Erklärung zu
einer generierten Meldung verwendet werden.

Maintenance	
title:	Name des MLM-Moduls
filename:	Name der Datei
version:	Version
institution:	Entwicklungsort
author:	Entwickler
specialist:	Verantwortliche Person vor Ort
date:	Datum der Inbetriebnahme
validation:	Status des MLMs
Library	
purpose:	Zweck des Moduls
explanation:	Erklärung der Arbeitsweise des Moduls
keywords:	Schlüsselworte
citations:	Literaturstellen
links:	Beziehungen zu anderen Informationsquellen
Knowledge	
type:	Art der MLM-Anwendung
data:	Benutzte Terme (Daten) im MLM-Modul
evoke:	Auslösung des Logic-Slot des Moduls
priority:	Priorität des Moduls
logic:	prozedurales Wissen im MLM-Modul
action:	Aktion des MLM-Moduls

Abb. 3: Übersicht über die Slots eines MLMs nach [Pryor 1991]

3.4.1.2. Die Knowledge Category

Die Knowledge Category enthält das das Verhalten des MLMs bestimmende Wissen. Der *Data-Slot* gibt an, welche Daten während der Verarbeitung des Wissens dem MLM zur Verfügung stehen müssen und wie auf diese zurückgegriffen werden kann. Seine Angaben sind im Gegensatz zu den anderen Slots in hohem Maße institutionsspezifisch, da die Arden-Syntax kein Datenreferenzmodell, welches ein medizinisches Informationsmodell und Zugriffsmethoden festlegt, verwendet. Ein typischer Eintrag in einem Data-Slot innerhalb einer Umgebung mit SQL als Datenbankabfragesprache ist

```
name := read {select name from patient where pat_id = 1500};
```

Hier wird der Variable *name* der Name des Patienten mit der Systemidentifikation 1500 zugewiesen. Institutionsspezifische Anteile, wie hier die Datenbankabfrage, sind in einem Medical Logic Moduls grundsätzlich daran zu erkennen, daß sie von geschweiften Klammern (engl. Curly braces) eingeschlossen sind. Sie sind nicht Bestandteil der Arden-Syntax, sondern müssen im gegebenen Anwendungsumfeld festgelegt werden. Je nach Umfang der Unterschiede der Formulierungsmöglichkeiten zwischen Institutionen können bei einem Austausch von MLMs kleinere oder größere Anpassungen notwendig werden. Innerhalb des Data-Slots erlaubt die Arden-Syntax weiterhin auch Operatoren, die die Aggregation von eventuell mehreren Werten einer Datenbankabfrage erlauben (z.B. AVERAGE, MINIMUM oder SUM) oder die zeitliche Einschränkungen bzgl. der Wertebereiche vornehmen (z.B. OCCURS WITHIN 2 DAYS PRECEDING, 4 MONTHS AGO oder 1 WEEK BEFORE NOW). Die Arden-Syntax sieht zu jedem aus der Datenbank gelesenen Wert das Vorhandensein einer sogenannten PRIMARY TIME vor. Diese Zeitangabe beinhaltet die „most medically relevant time for the value“ (Originalwortlaut aus [Hripcsak 1994]). Für den Wert eines Bluttests könnte dies beispielsweise der Zeitpunkt der Blutabnahme sein. Die genaue Bedeutung der PRIMARY TIME muß in der das entsprechende Medical Logic Modul einsetzenden Institution definiert werden.

Im *Evoke-Slot* sind die Ereignisse festgelegt, die zu einer Aktivierung des MLMs führen sollen. Die Auslösung kann direkt, nach einer definierten Verzögerung nach Auftreten eines Ereignisses oder aus einem anderen MLM oder Anwendungsprogramm erfolgen.

Der *Logic-Slot* beinhaltet die zu überprüfenden Entscheidungsregeln. Ein beispielhafter Eintrag könnte folgendermaßen aussehen :

```
if HEMATOCRIT is not number then
    conclude false;
endif;
if HEMATOCRIT <= PREVIOUS_HCT - 5 or HEMATOCRIT < 30 then
    conclude true;
endif;
```

Der Inhalt des Logic-Slots ähnelt klassischen prozeduralen Programmiersprachen wie bspw. Basic oder Pascal. Im ersten If-Statement wird sichergestellt, daß HEMATOCRIT einen gültigen numerischen Wert hat. Die Anweisung `conclude false` bewirkt einen sofortigen Abbruch der Verarbeitung des Medical Logic Modules. Im zweiten If-Statement ist das eigentliche medizinische Wissen, daß sowohl ein schnell fallender als auch ein Hämatokritwert unter 30 mit hoher Wahrscheinlichkeit auf eine kritische Entwicklung hinweisen, formuliert. In diesem Fall bewirkt die Anweisung `conclude true` die Ausführung der im *Action-Slot* enthaltenen Anweisungen. Typische Aktion des Action-Slots ist die Generation einer Meldung für einen Benutzer. Prinzipiell besteht aber auch die Möglichkeit, MLMs miteinander zu verketten, indem als Aktion ein anderes MLM angestoßen wird.

3.4.2. Eignung der Arden-Syntax zum Knowledge Sharing

Eines der Hauptziele beim Design der Arden-Syntax war ein möglichst unproblematischer Austausch von MLMs zwischen unterschiedlichen Institutionen. Ein solcher Austausch wird als Knowledge Sharing bezeichnet. Eine Folge dieses Designziels ist die Strukturierung der Wissensbank in voneinander unabhängige MLMs, was dazu führt, daß jedes MLM das *gesamte* nötige Wissen enthält, um *eine* konkrete wissensverarbeitende Funktion durchführen zu können. Praktische Untersuchungen [Pryor 1994] haben jedoch gezeigt, daß ein Austausch von MLMs zwischen unterschiedlichen Institutionen nicht ohne manuelle Anpassungen möglich ist.

Hripcsak stellt in [Hripcsak 1995] fest, daß Unterschiede im Vokabular und im Datenbankmodell für einen Großteil der nötigen Modifikationen im Rahmen des Knowledge Sharings verantwortlich sind. Es wird darauf verwiesen, daß durch Standardisierungen in diesem Bereich, der Modifikationsaufwand reduziert werden könnte. Die nötigen Änderungen können sich auf den institutionsspezifischen Teil innerhalb des Data-Slots beschränken, jedoch unter Umständen auch größere Modifikationen innerhalb des Logic-Slots nach sich ziehen.

Prokosch nennt in [Prokosch 1991] über Vokabular- und Datenmodellunterschiede hinausgehende Gründe, die ein Knowledge Sharing erschweren. Die Arden-Syntax erlaubt es, Wissen, das eigentlich in die Logic-Kategorie gehört, auf unterschiedliche Slots (insbesondere den Data- und Evoke-Slot) zu verteilen. So ist es beispielsweise möglich, im Data-Slot innerhalb oder auch außerhalb des institutionsspezifischen Abfrageteils temporale Einschränkungen oder auch Aggregationsoperatoren einzusetzen. Durch eine uneinheitliche Benutzung der Möglichkeiten kann es hier leicht zu Mißverständnissen innerhalb der empfangenden Institution kommen. Weiterhin kann Knowledge Sharing dadurch erschwert werden, daß ein MLM mit institutionsspezifischem Wissen überfrachtet wird. Dies ist oft dann der Fall, wenn ein MLM-Autor versucht, ein MLM möglichst gut in die vorhandene Umgebung des Informationssystems einzubetten oder lokale Datenbankabfragen zu optimieren.

3.4.3. Eignung der Arden-Syntax zum Knowledge-Reusing

Shwe führt ein weiteres Kriterium zur Beurteilung der Arden-Syntax ein, das eng mit dem Begriff des Knowledge Sharings verbunden ist: nämlich den des Knowledge Reusing [Shwe 1993]. Hierunter wird die Möglichkeit verstanden, Wissen in unterschiedlichen Problembereichen zu verwenden. Durch diese Möglichkeit würde eine redundante Haltung von Wissens-elementen überflüssig, was positive Auswirkungen auf die Wartbarkeit der Wissensbank zur Folge hätte. Shwe stellt hierzu fest, daß durch die prozedurale Wissensrepräsentation im Logic-Slots und der damit einhergehenden Vermischung von Fachwissen und Problemlösungswissen eine Wiederverwendbarkeit des Fachwissens in anderen Problembereichen sehr erschwert wird. Einen Ansatz zur Wiederverwendung von medizinischen Fachwissen zeigen [Linnarson 1989] und [Prokosch 1991], die vorschlagen, das medizinische Fachwissen in einem semantischen Netz zu modellieren und den Wissensmodulen als Infrastruktur zur Verfügung zu stellen. Ein solcher Ansatz erlaubt zwar ein auf Faktenwissen beschränktes Knowledge Reusing, erschwert jedoch Knowledge Sharing, da neben einzelnen MLMs nun noch das separat repräsentierte Fachwissen weitergegeben werden muß.

3.4.4. Beispiele für Arden-Syntax-basierte Systeme

Bisherige Arden-Syntax-basierte Systeme wurden in erster Linie im universitären Umfeld erstellt. Es wird erwartet, daß in den nächsten Jahren kommerzielle Krankenhausinformationssysteme um Arden-Syntax-basierte Subsysteme erweitert werden.

3.4.4.1. Linköping University

Das Linköping-University-MLM-Projekt hat Arden-Syntax-basierte entscheidungsunterstützende Systeme für eine patientenbetreuende Einrichtung, ein klinisches Labor und eine Forschungseinrichtung entwickelt. Sämtliche Systeme lassen sich in drei Komponenten zerlegen: in einen MLM-Editor, einen MLM-Compiler und eine sogenannte MLM-Engine. Gao beschreibt in [Gao 1992] weitere Details zur Systemarchitektur.

Es wurden Erfahrung hinsichtlich der Implementierung auf unterschiedlichen Rechnerplattformen und Datenbanksystemen gewonnen. Zentrale Anwendungsbereiche in der klinischen Einrichtung sind die Qualitätskontrolle von Laborergebnissen und die Unterstützung des Medikationsvorganges.

Es wurden weiterhin Methoden erarbeitet und Software entwickelt, die es anderen Systementwicklern erleichtern sollen, ähnliche Entscheidungssysteme zu realisieren. Angaben hierzu können in [Bang 1997] nachgelesen werden.

3.4.4.2. Columbia Presbyterian Medical Center

Bei dem Entwurf des dort entworfenen Arden-Syntax-basierten entscheidungsunterstützenden Systems wurde vor allem Wert auf eine gute Performance gelegt. Dies wurde durch eine spezielle Architektur des Gesamtsystems und einer Optimierung der Datenbankzugriffe erreicht. Die Grundlage hierfür war die Feststellung, daß ein Großteil des Zeitaufwandes für die Datenbankzugriffe benötigt wird, während der Zeitaufwand zur sonstigen Abarbeitung eines MLMs vernachlässigbar ist. Momentan sind circa fünfzig MLMs im Einsatz. Sie generieren täglich im Schnitt zwanzig klinische Warnungen, 2000 klinische Interpretationen, fünfzehn Forschungsnachrichten, zehn Meldungen zur Qualitätssicherung und 700 administrative Management-Nachrichten. Eine Untersuchung über die Akzeptanz ergab, daß der Großteil der Meldungen als sinnvoll erachtet wurde. Weitere Details hierzu können in [Hripcsak 1992] nachgelesen werden.

3.4.4.3. Institut für Medizinische Informatik in Gießen

Hier wurde bisher die Arden-Syntax zum Entscheidungsmonitoring im Bereich der Arzneimitteltherapie innerhalb eines pharmakologischen Systems eingesetzt [Prokosch 1991]. Konkret wird das behandelnde Personal bei der Erkennung von ungünstigen Arzneimittelinteraktionen sowie Allergiekontraindikationen unterstützt. Eine Besonderheit des Systems ist die Ablage von Fachwissen in deklarativer Form in einem semantischen Netz, auf welches bei der Problemlösung von einem MLM aus zugegriffen wird. Es hat sich herausgestellt, daß die Arden-Syntax zur Repräsentation des Wissens in der Pharmakologie gut geeignet ist.

4. Shell für wissensverarbeitende Funktionen für das GTDS

4.1. Anforderungen an das zu realisierende wissensbasierte System

Das GTDS soll um eine Umgebung erweitert werden, in der die Formulierung wissensverarbeitender Funktionen möglich ist. Zur Erreichung einer klinischen Einsetzbarkeit der resultierenden Funktionen muß bei der Systemkonzeption auf die Interessen aller bei einem klinischen Einsatz betroffenen Personengruppen Rücksicht genommen werden. Hierbei müssen sowohl die Interessen des GTDS-Anwenders, des Wissensverwalters als auch des GTDS-Systemverwalters berücksichtigt werden. Konkret sind dies folgende Anforderungen :

1. Die wissensverarbeitenden Funktionen sollen *in ein existierendes Informationssystem integriert* sein. Mit Integriertheit ist in diesem Zusammenhang gemeint, daß automatisch auf schon im Informationssystem vorhandene Daten zurückgegriffen wird („Datenintegriertheit“), daß die Problemlösung automatisch nach Auftreten eines Ereignisses im zugrundeliegenden Informationssystem ausgelöst wird („Ablaufintegriertheit“) und daß eventuelle durch die wissensverarbeitenden Funktionen generierte Meldungen den Benutzern automatisch und unmittelbar präsentiert werden. Hierdurch wird erreicht, daß der Anwender von wissensverarbeitenden Funktionen profitiert, ohne selbst einen zusätzlichen Arbeitsaufwand erbringen zu müssen oder seinen gewohnten Arbeitsablauf zu ändern. Wissensverarbeitende Funktionen, die diese Anforderungen erfüllen, werden in dieser Arbeit auch als integrierte wissensverarbeitende Funktionen bezeichnet.
2. Die *Wartung der integrierten wissensverarbeitenden Funktionen soll relativ problemlos sein*. Dies soll durch eine Strukturierung der Wissensbasis in voneinander unabhängige Wissensmodule, wobei jedes Modul das gesamte nötige Wissen zur Durchführung einer integrierten wissensverarbeitenden Funktion enthält, erreicht werden. Durch Löschen, Hinzufügen oder Modifizieren von Funktionen können so andere Funktionen nicht in ihrer Funktionsweise beeinträchtigt werden. Weiterhin müssen bei der Verifikation einer Funktion keine unüberschaubaren Abhängigkeiten zu anderen Modulen berücksichtigt werden.

3. Die Funktionsweise des Systems soll für die Anwender *nachvollziehbar* sein. Dies wird dadurch erreicht, daß jede für die Anwender sichtbare Aktivität des Systems auf genau eine Wissenskomponente zurückgeführt werden kann, die in einem leicht zu erlernenden und standardisierten Formalismus beschrieben ist. Diese Eigenschaft ist wichtig, damit der Anwender eigene Anpassungen selbst realisieren oder zumindest anregen kann.
4. Der dem System zugrundeliegende Formalismus zur Wissensrepräsentation soll *den Wissensaustausch* zwischen unterschiedlichen Institutionen nach Möglichkeit unterstützen. Dies kann zu einer Überwindung des von Feigenbaum beschriebenen Flaschenhalses der Wissensaquisition für wissensbasierte Systeme in der Medizin führen. Gerade im vorhandenen Umfeld - das GTDS wird an mehr als dreißig Zentren eingesetzt - könnte dies zu Synergieeffekten führen.
5. Das wissensbasierte System soll letztendlich eine ans GTDS gekoppelte *Shell für wissensverarbeitende Funktionen* darstellen. Als Shell wird hier eine Softwareumgebung bezeichnet, welche die Realisierung wissensverarbeitender Funktionen ermöglicht, aber selber noch kein diese Funktionen definierendes Wissen beinhaltet. Das Füllen der Shell mit Wissen entspricht dem Aufbau einer Wissensbank aus Wissensmodulen. Hierdurch soll eine flexible Berücksichtigung relevanter Anwendungsbereiche möglich werden.
6. Als Formalismus für die Wissensrepräsentation soll die *Arden-Syntax* verwendet werden. Sie scheint geeignet, die bisher definierten Anforderungen zu erfüllen. Die zur Erreichung der Operationalität eines Medical Logic Moduls notwendige Umwandlung in eine interne Repräsentation soll automatisch erfolgen.
7. Die Realisierung des Systems soll *zu möglichst geringfügigen Modifikationen am vorhandenen Informationssystem* - dem GTDS - führen. Diese physische Unabhängigkeit der Shell vom Informationssystem steht nicht im Widerspruch zu der weiter oben geforderten Integration. Durch diese Software-technische Eigenschaft wird auf den Systemverwalter des zugrundeliegenden Informationssystems Rücksicht genommen und damit eine Einführung ins klinische Umfeld erst praktisch möglich.

8. Die Shell soll einen „*retract-Mechanismus*“ beinhalten. Dieser Mechanismus erkennt störende Meldungen im Pool der noch zu präsentierenden Meldungen und ist damit die Grundlage für deren Ausblendung. Störende Meldungen resultieren aus der automatischen Auslösung der wissensverarbeitenden Funktionen verbunden mit einer meist nicht synchronen Meldungspräsentation und haben damit nichts mit Fehlern in der Wissensbank zu tun. Sie können unterteilt werden in doppelte und veraltete Meldungen. Unter doppelten Meldungen sind Meldungen zu verstehen, die während einer Meldungspräsentation mehrfach vorliegen; veraltete Meldungen sind solche, die zum Zeitpunkt einer Meldungspräsentation bereits veraltet sind. Die Namensgebung orientiert sich an [Kuperman 1997], der einen ähnlichen Mechanismus im Brigham Integrated Computing System (BICS) beschreibt. Dieser Mechanismus ist ein wichtiger Faktor zur Erreichung einer guten Benutzerakzeptanz.

4.2. Auswahl der Software

Bei der Softwareauswahl zur Realisierung der Shell für wissensverarbeitende Funktionen wird auf die durch das GTDS vorgegebene Software sowie allgemein verfügbare Standardwerkzeuge zurückgegriffen.

Konkret werden folgende Software-Komponenten verwendet

- Oracle RDBMS Version 7 (und aufwärts)
- Oracle SQL*Forms 3.0 bzw. Oracle Developer/2000
- C-Compiler und PRO*C-Erweiterung („embedded Sql“)
- Lex / Yacc Werkzeuge zur Compilererstellung.

Als Betriebssystem wurde UNIX gewählt, wobei jedes andere multitaskingfähige Betriebssystem ebenfalls geeignet ist, wenn es dafür die obengenannten Software-Komponenten gibt.

4.2.1. Oracle RDBMS Version 7

Beim *Oracle RDBMS (Version 7 und aufwärts)* handelt es sich um ein relationales Datenbankmanagementsystem, welches innerhalb des GTDS die zentrale datenverwaltende Rolle übernimmt. Besondere über die typischen Datenbankeigenschaften hinausgehende Merkmale von Version 7 sind die Verfügbarkeit von besonderen Datenbankobjekten wie Datenbanktriggern, Datenbankprozeduren und Prozedurpaketen.

4.2.1.1. Datenbanktrigger

Trigger sind eine spezielle Form von Prozeduren. Sie werden nicht durch expliziten Aufruf, sondern durch Auftreten eines datenbankspezifischen Ereignisses aktiviert. Diese datenbankspezifischen Ereignisse können anhand der drei Dimensionen

- Auslöseereignis,
- Auslösezeitpunkt und
- Triggertyp

weiter beschrieben werden.

Unter *Auslöseereignis* versteht man in diesem Zusammenhang den Typ des SQL-Befehls („insert“, „update“, „delete“), dessen Verwendung zur Auslösung des Triggers geführt hat. Der *Auslösezeitpunkt* („before“, „after“) gibt an, ob der Trigger vor oder nach dem Auslöseereignis ausgeführt werden soll. Der Auslösezeitpunkt „before“ erlaubt es, Spaltenwerte zu

manipulieren, während beim Auslösezeitpunkt „after“ sichergestellt ist, daß durch den Trigger keine Wertemodifikationen durchgeführt werden können. Der *Triggertyp* gibt schließlich an, ob der Trigger genau einmal pro auslösendem SQL-Befehl („befehlsorientiert“) oder für jeden durch den SQL-Befehl berührten Satz („datensatzorientiert“) ausgeführt werden soll. Durch Kombination dieser drei Dimensionen kann man theoretisch zwölf sich in der Art der Aktivierung unterscheidende Trigger konstruieren.

```
1  create or replace trigger IN_TUMOR
2  after INSERT on TUMOR
3  for each row
4  declare
5      s1 varchar2(50);
6      s2 varchar2(50);
7      s3 varchar2(50);
8      s4 varchar2(50);
9      s5 varchar2(50);
10 begin
11     s1 := :new.fk_patientpat_id;
12     s2 := :new.tumor_id;
13     eintrag_aufgetretene_events('INSERT_TUMOR',
14     s1,s2,s3,s4,s5);
14 exception when others then null;
15 end;
16 /
```

Abb. 4 : Beispiel für einen Datenbanktrigger

Die Abbildung 4 zeigt die Definition des Triggers IN_TUMOR, der ein möglicher Bestandteil einer konkreten Arden-Shell sein kann. Er hat die Aufgabe, nach einer durch einen Insert-Befehl verursachten Modifikation an der Tabelle Tumor, diesen Sachverhalt (einschließlich der Schlüsselwerte des betroffenen Satzes) zu protokollieren. Hierzu werden in den Zeilen 2 und 3 Auslösezeitpunkt, Auslöseereignis, kontrollierte Tabelle und der Triggertyp festgelegt. In den Zeilen 4 bis 15 werden die Aktivitäten des Triggers in der Oracle-typischen Sprache PL/SQL definiert. PL/SQL verknüpft die Verwendung von SQL mit den typischen Möglichkeiten einer prozeduralen Programmiersprache, wie bspw. Schleifen und Verzweigungen. Die Anweisung in Zeile 14 wird im Falle eines Fehlers ausgeführt und bewirkt, daß Fehlersituationen im Trigger nicht zu einem Scheitern der Gesamttransaktion führen.

Bei der Verwendung von Triggern muß die *Mutating-Table-Restriktion* berücksichtigt werden. Die Restriktion besteht darin, daß Trigger nicht mittels SQL-Befehlen auf die auslösende Tabelle zugreifen können. Technischer Hintergrund für diese Lese-Einschränkung ist, daß die Ausführung des Triggers Bestandteil der Transaktion ist, die durch die Manipulation der dem

Trigger zugeordneten Tabelle eingeleitet wurde und daß zur Sicherung der Lesekonsistenz zu Beginn einer Transaktion die jeweilige Tabelle exklusiv gesperrt wird, wodurch keine lesenden und schreibenden Zugriffe - natürlich auch nicht aus einem ausgelösten Trigger heraus - möglich sind. Das Wort „Mutating“ bezeichnet den Zustand des Gesperrtseins einer Tabelle während einer Transaktion. Ein „Schlupfloch“ zur Mutating-Table-Restriktion für einen Trigger des Typs „datensatzorientiert“ besteht in der Verwendung der Korrelationsnamen :old^a und :new, die es erlauben, innerhalb des PL/SQL-Codes des Triggers auf die einzelnen Felder des entsprechenden Datensatzes zuzugreifen. In den Zeilen 11 und 12 des obigen Beispiels wird mit dieser Methode auf das gerade eingefügte Feld `fk_patientpat_id` und `tumor_id` zugegriffen.

4.2.1.2. Datenbankprozeduren und Prozedurpakete

Bei Datenbankprozeduren und Prozedurpaketen handelt es sich um datenbankresidente PL/SQL-Programme, d. h. um Programme, die nicht, wie sonst oft üblich, in der Anwendung lokalisiert sind. Der Unterschied zwischen Datenbankprozeduren und Prozedurpaketen liegt lediglich darin, daß es sich bei letzteren um eine Gruppe von Prozeduren handelt. Eine Besonderheit der Paketprozeduren ist die Möglichkeit der *Überladung von Prozedurnamen*. Diese Möglichkeit wird im englischen Sprachgebrauch als „Overloading“ bezeichnet. Dabei werden innerhalb der Schnittstellendefinition eines Paketes mehrere Prozeduren mit identischem Namen definiert, die sich jedoch in der Parameterliste, d.h. bezüglich der Anzahl der Parameter und/oder den Typen der Parameter, unterscheiden. Zu diesen unterschiedlichen Schnittstellendefinitionen, die jedoch alle den gleichen Namen besitzen, werden die unterschiedlichen Prozeduren später innerhalb des Paketes implementiert. Typische Kandidaten für eine Namens-Überladung sind Prozeduren, die sinnvoll mit unterschiedlichen Parametern, wie dies bei Konvertierungsfunktionen typischerweise der Fall ist, aufgerufen werden können. Wird eine überladene Prozedur aufgerufen, dann wird auf Grund der Parameterliste zur Laufzeit entschieden, welche der ausprogrammierten Prozeduren des Paketes tatsächlich zur Ausführung kommt. Als Beispiel kann der in Abbildung 12 gezeigte Ausschnitt aus dem Prozedurpaket `ARDEN_PACK` herangezogen werden.

^a alle Namen für SW-Objekte, wie beispielsweise Programme, Prozeduren oder Variablen, sowie Programmcode sind durch diesen Schrifttyp optisch gekennzeichnet

Von den Möglichkeiten dieser neuen Datenbankobjekte wird Gebrauch gemacht, um eine weitestgehende physische Unabhängigkeit der Shell für wissensverarbeitende Funktionen von der eigentlichen Anwendung des GTDS zu erreichen.

4.2.2. Oracle-Tools : Oracle SQL*Forms und PRO*C

Bei *Oracle SQL*Forms* handelt es sich um ein produktives Entwicklungswerkzeug zur Erstellung von masken- und menübasierten Anwendungen. Der *Developer/2000* ist der Nachfolger von SQL*Forms. Während mit SQL*Forms nur zeichenorientierte Anwendungen erzeugt werden können, erlaubt der Developer/2000 die Erstellung benutzerfreundlicher graphischer Oberflächen. Die Verwendung dieser Werkzeuge ergibt sich aus der Notwendigkeit, daß die Shell in irgendeiner Form eine Schnittstelle zum Benutzer, unabhängig ob es sich um den Wissensverwalter oder den eigentlichen Anwender handelt, besitzen muß.

*PRO*C* ermöglicht den Zugriff auf bzw. die Manipulation einer Oracle-Datenbank mittels SQL-Befehlen aus einem C-Programm heraus. Die Anwendung dieser Methode kann manchmal notwendig werden, obwohl die Oracle-Werkzeuge, insbesondere das obengenannte SQL*Forms, sehr vielfältig und leistungsstark sind.

4.2.3. Tools zur Compilerentwicklung : Lex und Yacc

Bei *Lex* und *Yacc* handelt es sich um Werkzeuge zur Compilererstellung, die seit Anfang der siebziger Jahre fester Bestandteil des Betriebssystems Unix sind, aber auch auf Windows NT basierten Rechner unter dem Namen *Flex* und *Bison* verfügbar sind. Ein Compiler ist hierbei allgemein ein Programm, welches der Übersetzung einer Sprache in eine andere dient. Hierbei deckt *Lex* die lexikalische Analyse ab, während *Yacc* die späteren Phasen - die syntaktische Analyse, die semantische Analyse sowie die Codeerzeugung - unterstützt.

Die Notwendigkeit eines Compilers für die Shell für wissensverarbeitende Funktionen ergibt sich daraus, daß die Arden-Syntax als verwendeter Formalismus zur Wissensrepräsentation zur Erreichung einer Operationalität in eine in der Softwareumgebung ablauffähige interne Repräsentation umgeformt werden muß (vgl. Kapitel 4.4.2).

Weitere Informationen über die oraclespezifischen Softwarekomponenten können in [Stürner 1993], [Hein 1991] sowie [Froese 1994], über die Compilerwerkzeuge `Lex` und `Yacc` in [Levine 1994] und [Aho 1990] nachgelesen werden.

4.3. Notwendige Ergänzungen der Arden-Syntax

In diesem Kapitel sollen die Abweichungen der implementierten Syntax zur Beschreibung der MLMs von der im Standard definierten Arden-Syntax dargestellt werden. Es gibt hierbei Abweichungen, die den Sprachumfang innerhalb des Data-Slots sowie innerhalb des Logic-Slots betreffen.

- Innerhalb des Data-Slots überläßt die Arden-Syntax bei den Variablendefinitionen und Eventdefinitionen dem Entwickler eines Arden-basierten Systems die Festlegung einer an die vorhandenen Gegebenheiten angepaßten Syntax.
- Innerhalb des Logic-Slots wurden einige Modifikationen am Sprachumfang vorgenommen. Konkret wurden zum einen einige der vorgesehenen Formulierungsmöglichkeiten weggelassen, da keine sinnvolle Verwendung im Rahmen des GTDS möglich ist. Einige andere Arden-Syntax-fremde Formulierungen waren erforderlich und wurden in den Sprachumfang aufgenommen.

Einzelheiten werden in den folgenden Unterkapiteln genannt.

4.3.1. Ergänzungen im Data-Slot

Im Data-Slot der Knowledge-Kategorie ist die Definition von Variablen einschließlich einer Wertzuweisung aus den Daten des Informationssystems sowie von Ereignissen vorgesehen. Abbildung 5 zeigt einen vereinfachten Ausschnitt der Grammatikregeln der Arden-Syntax zur Festlegung der Formulierungsmöglichkeiten im Data-Slot.

Zunächst einige Anmerkungen zur Schreibweise der folgenden Grammatikregeln: In der Darstellung ist der Kopf einer Regel mit „:=“ vom Körper der Regel getrennt. Das Zeichen „|“ erlaubt die Aufzählung von unterschiedlichen Alternativen innerhalb eines Regelkörpers. Terminalsymbole - also Symbole die direkt aus dem Eingabe gelesen werden - sind in Großbuchstaben geschrieben und von Anführungszeichen eingeschlossen. Nichtterminalsymbole - also Symbole die für eine syntaktisch korrekte Kombination von Terminalsymbole stehen - sind von „>“ und „<“-Zeichen eingeschlossen.

```

<data_assignment> ::=
    <identifier> ":" <data_assign_phrase>

<data_assign_phrase> ::=
    "READ" <data_factor>
    | "READ" <of_read_func_op> <data_factor>
    | "READ" <from_of_func_op> <data_factor>
    | "EVENT" <event_factor>

<of_read_func_op> ::=
    "AVERAGE"
    | "COUNT"
    | "SUM"

<from_of_func_op> ::=
    "MINIMUM"
    | "MAXIMUM"
    | "LAST"
    | "FIRST"
<data_factor> ::=
    "{" <data_mapping> "}"
<event_factor> ::=
    "{" <event_mapping> "}"

```

Abb.5: Vereinfachte Darstellung einiger Grammatikregeln des Data-Slots

Die erste Regel führt das Symbol `<data_assignment>` ein, welches für eine korrekte Variablen- oder Ereignisdefinition steht. Aus dem Regelkörper ist entnehmbar, daß eine solche Definition aus einem `<identifier>` - einem Ereignis-bzw. Variablennamen - und einem ihm zugewiesenen Konstrukt `<data_assign_phrase>` besteht. Der syntaktische Aufbau der `<data_assign_phrase>` ist in den folgenden Regeln beschrieben. Wesentlich ist, daß der Sprachumfang der Arden-Syntax lediglich einrichtungsunabhängige Konstruktionen, wie den prinzipiellen Aufbau der Zuweisung oder mögliche Operationen auf den gelesenen Werten, abdeckt. An der Stelle, wo die einrichtungsspezifische Syntaxbeschreibung der Datenabfrage oder Ereignisdefinition käme, enden die Vorgaben der Arden-Syntax: es wird lediglich ein in geschweiften Klammern eingeschlossenes Nichtterminalsymbol `<data_mapping>` bzw. `<event_mapping>` definiert. Ein aktueller Forschungsschwerpunkt im Umfeld der Arden-Syntax ist die Inkorporierung von einheitlichen Datenreferenzmodellen (einschließlich Abfragemethodik). In diesem Falle könnte an dieser Stelle eine weitere Syntaxspezifikation erfolgen.

```

<data_mapping> ::=
    SQL ":" <select_statement>

<select_statement> ::=
    SELECT <opt_all_distinct> <selection_einfach> <table_exp>

<event_mapping> ::=
    <ev_operation> "," <table> "," IDENTIFIER "," STRING
    | <ev_operation> "," <table> "," IDENTIFIER
    | <ev_operation> "," <table>

```

Abb. 6: Hinzugefügte Regeln zur Definition der Symbole `event_mapping` und `data_mapping`

Im Rahmen der vorliegenden Arbeit wurden die Symbole `<data_mapping>` und `<event_mapping>`, wie aus Abbildung 6 ersichtlich, definiert. Aus Gründen der Übersichtlichkeit wurde im Körper der ersten Regel auf die Darstellung weiterer syntaktischer Alternativen verzichtet. Die folgenden Unterkapitel verdeutlichen die Grammatikregeln anhand einiger gültiger Variablen- und Ereignisdefinitionen.

4.3.1.1. Mögliche Variablendefinitionen

Basierend auf den Grammatikregeln aus Abbildung 6 sind die in Abbildung 7 gezeigten Variablendefinitionen möglich.

```
HIST_LISTE_D := READ {SQL: select fk_histologie_shis from  
histologie where fk_tumorfk_patient = iarc6.fk_tumorfk_patient  
and fk_tumortumor_id = iarc6.fk_tumortumor_id };
```

```
HIST_LISTE_D_TEXT := READ {SQL: select htext from histologie  
where fk_tumorfk_patient = iarc6.fk_tumorfk_patient and  
fk_tumortumor_id = iarc6.fk_tumortumor_id };
```

```
PAT_NAME := READ last{SQL: select name from patient where  
pat_id = iarc6.fk_tumorfk_patient};
```

```
PAT_GESCHL := READ last{SQL: select geschlecht from patient  
where pat_id = iarc6.fk_tumorfk_patient};
```

Abb. 7: Beispiele für gültige Variablenzuweisungen

In der Abbildung 7 werden vier Variablen Werte zugewiesen. Hierbei stellen die ersten beiden Variablen Listen dar, in denen sämtliche Histologieschlüssel bzw. die entsprechenden Histologiebeschreibungen zu einem Tumorfall enthalten sind. Da die Arden-Syntax keine explizite Typisierung von Variablen vorsieht, wird der Typ implizit aus dem Aufbau der Zuweisung geschlossen. Die Variable `HIST_LISTE_D_TEXT` beispielsweise muß vom Typ Liste sein, da keine Aggregationskommandos wie `MINIMUM` oder `AVERAGE` in der Zuweisung verwendet wurden. Die eigentliche Datenabfrage ist in SQL unter Verwendung der GTDS-Datenmodell spezifischen Tabellen- und Feldnamen formuliert. Da SQL eine Standardabfragesprache für relationale Datenbanken ist, ist eine leichte Nachvollziehbarkeit der Variablenzuweisungen, wie sie im Rahmen von Knowledge Sharing nötig ist, gewährleistet.

4.3.1.2. Mögliche Ereignisdefinitionen

Hintergrund der festgelegten Syntax ist der Wunsch, jede mögliche Veränderung im Datenbestand als ein Ereignis definieren zu können. Um dies zu erreichen, kann jedes Ereignis durch bis zu vier Angaben beschrieben werden. Die erste Angabe legt den Typ des die Änderung ausgelösten Datenmanipulationsbefehls fest. Es kann hier zwischen `insert` und `update` unterschieden werden. Die zweite Angabe legt die durch die Änderung betroffene Tabelle fest. Bei diesen ersten beiden Angaben handelt es sich um Mußangaben einer Ereignisdefinition. Durch weitere Angaben kann die Spezifität des Ereignisses weiter verstärkt werden. Durch die optionale dritte Angabe kann ein Feld der Tabelle festgelegt werden, das geändert werden muß, damit das Ereignis vorliegt. In einer vierten optionalen Angabe kann die Ereignisspezifität durch einen bestimmten Wert für dieses Feld weiter vergrößert werden. Basierend auf den Grammatikregeln aus Abbildung 6 sind die in Abbildung 8 gezeigten Ereignisdefinitionen möglich.

```
patient_neuaufnahme := EVENT {insert, patient}
insert_opdatum      := EVENT {insert, operation, opdatum}
insert_autopsie      := EVENT {insert, patient, autopsie, "J"}
UPDATE_HISTOLOGIEiarc6 := EVENT {update, histologie,
fk_histologie_shis}
```

Abb. 8: Beispiele für gültige Ereignisdefinitionen

Beim Ereignis `patient_neuaufnahme` handelt es sich beispielsweise um ein relativ unspezifisches Ereignis, welches nach Einfügen eines neuen Satzes in die Patiententabelle vorliegt. Das Ereignis `insert_autopsie` hingegen liegt nur vor, wenn zudem in das Feld Autopsie der Wert „J“ eingetragen wurde. Abbildung 9 stellt systematisch die sich ergebenden unterschiedlichen Ereignistypen mit steigender Spezifität übersichtlich dar.

Typ	Syntax	Beschreibung
1	EVENT {insert,<tabelle>}	liegt vor, wenn in der spezifizierten Tabelle ein Datensatz eingefügt wurde.
2	EVENT {update,<tabelle>}	liegt vor, wenn in der spezifizierten Tabelle ein Datensatz verändert wurde.
3	EVENT {insert,<tabelle>,<feld>}	wie Typ 1, zusätzlich muß im spezifizierten Feld irgendein Wert eingetragen worden sein. Der Wert dieses Feldes ist unerheblich, solange ein Wert vorliegt (Wert also ungleich NULL).
4	EVENT {update,<tabelle>,<feld>}	wie Typ 2, zusätzlich muß im spezifizierten Feld ein neuer Wert eingetragen worden sein. Der Wert dieses Feldes ist unerheblich, solange sich dieser vom vorherigen Wert unterscheidet.
5	EVENT {insert,<tabelle>,<feld>,<wert>}	wie Typ 3, zusätzlich muß der eingetragene Wert dem spezifizierten Wert entsprechen.
6	EVENT {update,<tabelle>,<feld>,<wert>}	wie Typ 4, zusätzlich muß der eingetragene Wert dem spezifizierten Wert entsprechen.

Abb. 9: Übersicht über die vorhandenen Ereignistypen

Im Kapitel 4.5. werden mögliche Auswirkungen einer zweckmäßigen Wahl der Ereignisspezifität auf das Laufzeitverhalten der Shell und den Inhalt des Logic-Slots diskutiert.

4.3.2. Modifikationen im Logic-Slot

4.3.2.1. Die Primary Time

Innerhalb der Arden-Syntax ist die Existenz einer sogenannten `PRIMARY TIME` zu jeder im Data-Slot definierten Variable vorgesehen, auf die im Logic-Slot zugegriffen werden kann. Die Bedeutung dieser Zeit ist jedoch nicht eindeutig definiert. Bei einer Variablen, die die Histologie eines Tumors enthält, ist es beispielsweise nicht klar, ob die Erhebungszeit, die Erfassungszeit oder das Diagnosedatum des Tumors für die `PRIMARY TIME` relevant ist. Man kann sich auf den Standpunkt stellen, daß diese Entscheidung durch eine Servicefunktion eines zugrundeliegenden Data Dictionaries erfolgen kann. Dies löst jedoch nur auf den ersten Blick das Problem und verschlechtert die Kontrolle des Wissensverwalters über das Verhalten einer wissensverarbeitenden Funktion. Man kann vermuten, daß die Autoren der Arden-Syntax von Datenerfassungssystemen - wie beispielsweise Laborwerteerfassungssystemen - inspiriert wurden, die jede Information automatisch mit einem Zeitstempel versehen. Da dies innerhalb des GTDS nicht der Fall ist, wurde bei der gemachten Realisierung keine automatische Generierung einer `PRIMARY TIME` vorgenommen.

4.3.2.2. Die SUBSTR-Funktion

Der dem Logic-Slot zugrundeliegende Sprachumfang wurde um eine `SUBSTR`-Funktion erweitert. Diese Funktion erlaubt das beliebige Herausschneiden eines Teils aus einer Zeichenkette. Diese Funktionalität erleichtert die Verwendung der in den meisten Schlüsselsystemen implizit in den Schlüsseln enthaltenen hierarchischen Beziehungen und ist für viele Anwendungen innerhalb des GTDS denkbar.

4.3.2.3. Die Variable MELDUNGS-MODUS

Eine weitere Erweiterung des Sprachumfangs innerhalb des Logic-Slots besteht in einer automatischen Zurverfügungstellung der Variablen `MELDUNGS-MODUS`, die einen der Werte Fehler, Information oder Warnung enthalten kann. Jede von MLMs generierte Meldung wird mit dem aktuellen Wert dieser Variablen gekennzeichnet. So wird es möglich, Meldungen dynamisch als Information, Warnung oder Fehlermeldung zu markieren und die Shell in die

Lage zu versetzen, Meldungen so zu kennzeichnen, daß auf den ersten Blick ihre Wichtigkeit erkannt werden kann.

4.4. Architektur der Shell für wissensverarbeitende Funktionen

Zur Integration wissensverarbeitender Funktionen in das Gießener Tumordokumentationssystem wurde die Shell für wissensverarbeitende Funktionen entwickelt. Hierbei wurde auf die Erfüllung der in Kapitel 4.1. formulierten Anforderungen zur Erreichung einer klinischen Einsetzbarkeit großer Wert gelegt. Die Abbildung 10 gibt eine erste Übersicht über die Systemarchitektur.

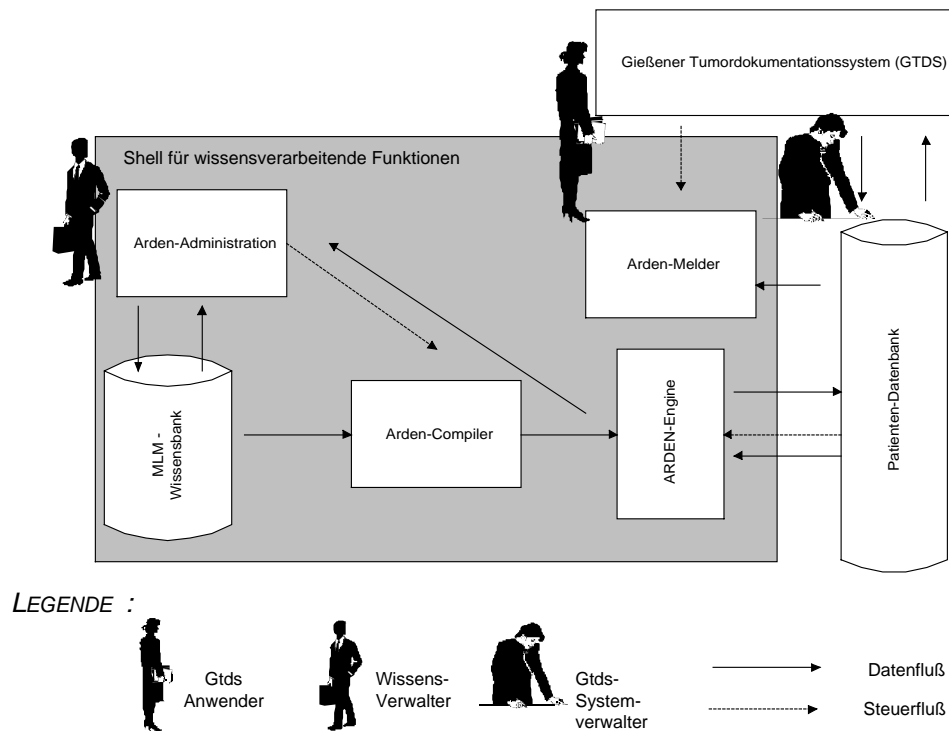


Abb. 10: Übersicht über die Architektur der Shell für wissensverarbeitende Funktionen

Die folgenden Unterkapitel erläutern die Funktion der einzelnen Komponenten und gehen auf ausgewählte Aspekte der Realisierung ein.

4.4.1. Die Arden-Administration

Diese Komponente ist die Steuerungszentrale für den Wissensverwalter und erlaubt ihm Funktionen zum Management der modularen Wissensbank sowie zur Kontrolle und zum Aufruf anderer Komponenten.

- *Import / Export von MLMs*: Die Shell legt die einzelnen MLMs innerhalb relationaler Tabellen der Datenbank ab. Will man ein vorhandenes MLM in einer anderen Institution einsetzen, so kann man dieses in eine Textdatei exportieren, die dann Gegenstand der

Übermittlung ist. Umgekehrt ist es möglich, über einen Importvorgang ein in einer Textdatei vorliegendes MLM in den vorgesehenen Tabellen abzulegen.

- *Editieren von MLMs*: der Wissensverwalter kann über diese Komponente neue MLMs erzeugen oder vorhandene MLMs verändern. Dieses ist sinnvoll, wenn eine neue konkrete wissensverarbeitende Funktion realisiert werden soll oder aber das Verhalten einer vorhandenen verändert werden soll. Weiterhin kann das Editieren eines MLMs nötig werden, wenn während des Umwandlungsvorganges in die interne Darstellung syntaktische Fehler im MLM festgestellt worden sind.
- *Initialisierung der Shell*: Bei Ausführung dieser Funktionalität wird der Inhalt von Systemtabellen, die zur Laufzeit der Shell (während des Entscheidungsmonitorings) mit Einträgen versehen werden, gelöscht. Die Bedeutung der Systemtabellen und ihrer Einträge wird bei der noch folgenden Beschreibung der Komponente Arden-Engine deutlich.
- *Nachvollziehen der Aktivität der Shell*: Hierbei werden die Systemtabellen, die die Aktivität der Shell widerspiegeln, übersichtlich dargestellt. Beispielsweise kann aus ihnen entnommen werden, welche Ereignisse aufgetreten sind, zu welchen MLM-Aufrufen diese Ereignisse geführt haben, welche Meldung die MLMs generiert haben und ob diese vom entsprechenden Benutzer bereits gelesen wurden. Weitere Details können in der Beschreibung der Arden-Engine nachgelesen werden.
- *Aufruf des Arden-Compilers*: Der externe Repräsentationsformalismus einer wissensverarbeitenden Funktion ist die Arden-Syntax. Zur Erreichung einer Operationalität des MLMs durch die Shell, muß das MLM in eine interne ausführbare Darstellung überführt werden. Diese Umsetzung von der externen in die interne Darstellung wird durch den Arden-Compiler ausgeführt, der Gegenstand des nächsten Kapitels ist.

4.4.2. Der Arden-Compiler

Der Arden-Compiler wandelt zur Erreichung einer Operationalität ein gegebenes MLM in eine interne Repräsentation um. Die Abbildung 11 zeigt schematisch die hierbei entstehenden Objekte.

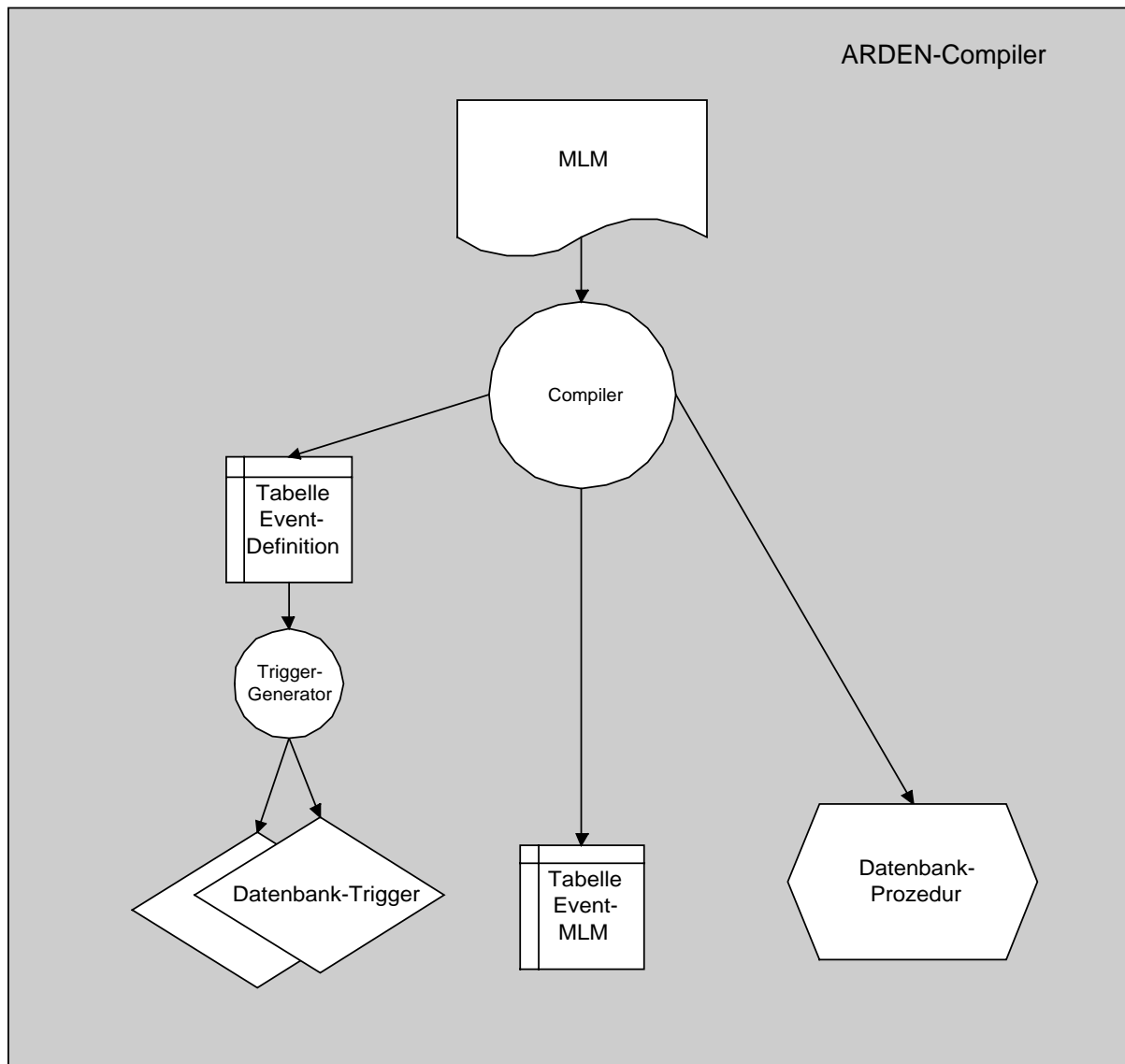


Abb.: 11 : Grobe Funktionsweise des Arden-Compilers

Die Umwandlung in die interne Darstellung geschieht in mehreren Schritten. Im ersten Schritt wird vom Compiler für das MLM eine Datenbankprozedur sowie Einträge in die Systemtabellen `ARDEN_EVENT_DEFINITION` und `ARDEN_EVENT_MLM` generiert. Die Datenbankprozedur spiegelt im wesentlichen den Data-Slot, den Logic-Slot und den Action-Slot des MLMs wieder. Die Tabelle `ARDEN_EVENT_DEFINITION` enthält zu dem/den Ereignis/sen des MLMs die Angaben, die in der Eventdefinition innerhalb des Data-Slots vorgenommen wurden. Der im Event-Slot definierte Auslösungsmodus eines MLMs, d.h., ob das MLM sofort oder mit einer definierten Verspätung nach Auftreten eines entsprechenden Ereignisses ausgeführt werden soll, wird in der Systemtabelle `ARDEN_EVENT_MLM` festgehalten.

Im einem zweiten Schritt werden ausgehend von den im ersten Schritt durchgeführten Einträgen in `ARDEN_EVENT_DEFINITION` vom Triggergenerator neue Datenbanktrigger

generiert bzw. eventuell vorhandene modifiziert, die der Erkennung relevanter Ereignisse innerhalb des GTDS dienen. Da es nicht beliebig viele Datenbanktrigger zu einer Datenbanktabelle geben kann, müssen durch einen Datenbanktrigger ggf. mehrere MLMs ausgelöst werden können. Die Konzeption sieht das Vorhandensein von max. zwei Datenbanktriggern - einen für das SQL-Statement Insert, einen für das SQL-Statement Update - pro Tabelle vor. Innerhalb des Triggerkörpers wird nach Analyse der innerhalb des Datenbanktriggers zur Verfügung stehenden Werte entschieden, welche / welches Ereignis aufgetreten ist.

Zur Realisierung des Arden-Compilers wurden die UNIX-Tools `yacc` und `lex` verwendet. Die Komplexität der Codeerzeugung wurde durch das Prozedurpaket `ARDEN_PACK` reduziert, das in der Laufzeitumgebung der vom Arden-Compiler erzeugten Datenbankprozedur zur Verfügung steht. Dieses Prozedurpaket stellt eine Sammlung von grundlegenden Funktionen und Prozeduren für Vergleiche, Zuweisungen und sonstige Operationen dar, bei deren Definition das Prinzip des Überladens von Prozedurnamen verwendet wurde. Der Arden-Compiler verwendet bei der Codeerzeugung nach Möglichkeit Prozeduren aus dieser Sammlung, wodurch für ihn eine Differenzierung nach den Typen der Argumenten entfällt. Soll beispielsweise innerhalb des MLMs der Inhalt einer Arden-Syntax-Variable an einen Textstring angehängt werden, so wird im generierten PL-SQL-Code immer die Prozedurpaket-Funktion `AUSGABE` verwendet. Innerhalb des Prozedurpaketes werden abhängig vom Variablentyp die richtigen Konvertierungen durchgeführt. Ohne die Verwendung der Funktion `AUSGABE` müßte der Arden-Compiler in Abhängigkeit vom Typ der anzuhängenden Variablen unterschiedlichen Code erzeugen.

Die Abbildung 12 zeigt einen Ausschnitt aus dem erstellten Prozedurpaket, in dem die Funktion `AUSGABE` definiert wurde. Man erkennt die grundsätzliche Zweiteilung eines Prozedurpaketes in einen Kopfteil („Schnittstelle“ des Paketes), in dem sämtliche Prozedur- und Funktionsnamen einschließlich der Parametertypen aufgelistet sind, und einen Körper, in dem die genannten Prozeduren und Funktionen ausprogrammiert sind. Konkret existieren vier Funktionen mit dem Namen `AUSGABE` für die Typen `varchar2`, `number`, `arden_pack.tp_tabelle` und `arden_pack.tp_tabelle_date`. Bei den ersten beiden handelt es sich um atomare Typen für Zeichenketten bzw. Zahlen, bei den letzten beiden um selbstdefinierte Typen. `arden_pack.tp_tabelle` erlaubt hierbei listenförmige Variablen, wobei die einzelnen Listenelemente einfache Zeichenketten sind.

arden_pack.tp_tabelle_date unterscheidet sich vom vorherigen Typ dadurch, daß es sich bei den Listenelementen um Datumsangaben handelt.

```
create or replace package arden_pack as
...
/*Ausgabe mit Overloading*/
function ausgabe(atom in varchar2)
    return varchar2;
function ausgabe(tabelle in arden_pack.tp_tabelle)
    return varchar2;
function ausgabe(tabelle in arden_pack.tp_tabelle_date)
    return varchar2;
function ausgabe(atom in number)
    return varchar2;
...
end arden_pack;

create or replace package body arden_pack as
...
/*Ausgabe mit Overloading*/
function ausgabe(atom in varchar2)
    return varchar2
is
    result varchar2(255);
begin
    result := atom;
    return result;
end ausgabe;
/****/
function ausgabe(atom in number)
    return varchar2
is
    result varchar2(255);
begin
    result := to_char(atom);
    return result;
end ausgabe;
/****/
function ausgabe(tabelle in arden_pack.tp_tabelle)
    return varchar2
is
    result varchar2(255);
begin
    result := arden_pack.ausgabe_tabelle_string(tabelle);
    return result;
end ausgabe;
/****/
...
end arden_pack;
```

Abb. 12 : Definition der überladenen Funktion AUSGABE im Prozedurpaket ARDEN_PACK

Im Anhang ist die komplette Schnittstelle des Paketes ARDEN_PACK enthalten.

4.4.3. Die Arden-Engine

Diese Komponente ist verantwortlich für das korrekte Laufzeitverhalten während des Entscheidungsmonitorings vom Auftreten eines Ereignisses bis zur eventuellen Generierung einer Meldung. Die Komponente besteht aus Datenbanktriggern, Datenbankprozeduren, Datenbanktabellen und einem C-Programm. Die Funktionsweise wird in Abbildung 13 schematisch dargestellt.

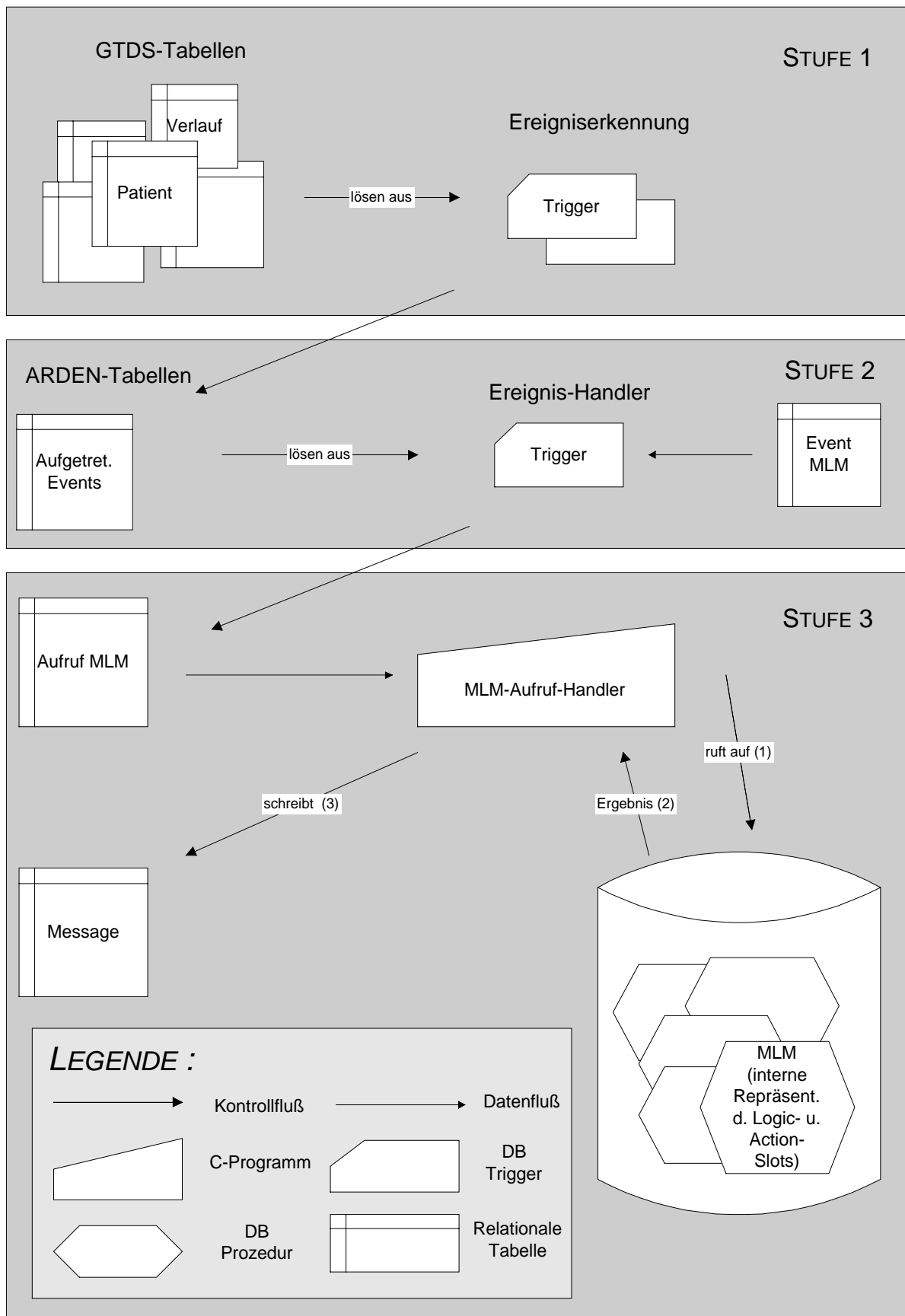


Abb. 13 : Bestandteile und Funktionsweise der Arden-Engine

Der Vorgang vom Erkennen eines Ereignisses bis zur eventuellen Generierung einer Meldung geschieht in drei Stufen :

1. Tritt im Anwendungsbereich eines Informationssystems ein Ereignis (z.B. eine bestimmte Medikation, Patientenneuaufnahme oder Dokumentation einer Tumorhistologie) auf, so resultiert daraus eine Veränderung im Datenbestand des Informationssystem. Handelt es sich bei dieser Veränderung um ein Ereignis, das für ein oder auch mehrere im System integrierte MLMs auslösenden Charakter hat, so wird der entsprechende Trigger aktiv und speichert den Namen des Ereignis sowie Eigenschaften wie die Uhrzeit, die Benutzerkennung des verursachenden Anwenders und die Schlüsselattribute/werte des ereignisauslösenden Eintrages in der Tabelle `ARDEN_AUFGETRETENE_EVENTS` ab. Es wird nicht schon an dieser Stelle ein MLM ausgelöst, da dieses aufgrund der Mutating-Table-Restriktion in seinen Aktionsfähigkeiten eingeschränkt wäre, und da die Arden-Syntax verschiedene Auslösungsmodi, wie „direct“, „delayed“ oder „periodical“ nach Auftritt eines Ereignisses, vorsieht.
2. Bei jedem Eintrag in die Tabelle `ARDEN_AUFGETRETENE_EVENTS` wird wiederum ein Trigger aktiv, der die abgespeicherten Werte liest, sich in der Tabelle `ARDEN_EVENT_MLM` über die mit dem Ereignis assoziierten MLMs und über deren Auslösungsmodus informiert und schließlich einen oder mehrere Einträge in der Tabelle `ARDEN_AUFRUF_MLM` erstellt. Jeder Eintrag enthält den Namen des aufzurufenden MLMs, die Zeit des Aufrufes und weitere Angaben, die später beim Aufruf des MLMs als Aufrufparameter verwendet werden.
3. Der Aufruf der MLMs wird basierend auf dieser Tabelle vom in der MLM-Aufruf-Handler durchgeführt. Bei diesem in C realisierten Programm handelt es sich um einen periodisch aktiven Hintergrundprozeß. Dieser ist notwendig, um die Ausführung der entsprechenden Datenbankprozedur aus der Transaktion, die mit dem durch den GTDS-Benutzer abgesetzten Statement zur Veränderung des Datenbestandes beginnt, herauszulösen. Würde man nicht so verfahren, so könnte es zu unakzeptablen Deadlock-Situationen kommen, wenn aus einer aufgerufenen Datenbankprozedur, die ja dann Bestandteil der Transaktion wäre, auf gesperrte Tabellen zugegriffen würde. Bei Aufruf eines MLMs werden diesem neben dem Zeitpunkt des Ereignisauftritts noch Kontextinformationen zum auslösenden Ereignis zur Verfügung gestellt. Mit dem Begriff Kontextinformationen sind hier die Schlüsselwerte des Datensatzes zu verstehen, dessen Eintrag bzw. Modifikation letztlich zum MLM-Aufruf geführt haben. Diese Kontextinformationen können vom MLM dazu

benutzt werden, beliebige für den Ableitungsvorgang notwendige Daten aus der Informationssystemdatenbank zu lesen. Die Architektur sieht weiterhin vor, daß eventuelle für den Anwender bestimmte Meldungen des MLMs in die Tabelle `ARDEN_MESSAGE` eingetragen werden. Zu jedem Meldungseintrag enthält sie neben dem eigentlichen Meldungstext

- Angaben, die ein Nachvollziehen des Ableitungsprozesses erlauben,
- die Benutzerkennung des Anwenders für den eine betrachtete Meldung bestimmt ist, sowie
- die Übermittlungsart („EMAIL“, „SCREEN“), die festlegt, auf welchem Weg die Nachricht den Adressaten erreichen soll.

Basierend auf der Tabelle `ARDEN_MESSAGE` präsentiert der Arden-Melder dem Benutzer die relevanten Meldungen

4.4.3.1. Der retract - Mechanismus

Der in Kapitel 4.1. geforderte retract-Mechanismus zur Ausblendung störender Meldung ist in den MLM-Aufruf-Handler der Arden-Engine integriert. Seine Funktionsweise ist verblüffend einfach und wird dadurch möglich, daß dem MLM-Aufruf-Handler neben dem Namen des aufzurufenden MLMs noch die zuvor beschriebenen Kontextinformationen zur Verfügung stehen. Jedesmal *bevor* der MLM-Aufruf-Handler ein MLM aufruft, kontrolliert er, ob bereits eine Meldung vom gleichen MLM zum identischen Kontext im Pool der noch zu präsentierenden Meldungen vorliegt. Ist dies der Fall, so wird diese Meldung als „störend“ gekennzeichnet. Diese Meldung wird dann vom Arden-Melder ignoriert, d.h. daß sie nicht dem Anwender präsentiert wird. Erst dann fährt der MLM-Aufruf-Handler mit dem Aufruf des aktuellen MLMs fort.

Besonderes Merkmal neben der Einfachheit der Realisierung ist, daß der Mechanismus allgemein für alle MLMs gilt und transparent für den Entwickler der MLMs ist.

4.4.4. Der Arden-Melder

Diese Komponente dient dem Zweck, die durch die MLM-Engine generierten Meldungen dem richtigen Benutzer zu präsentieren. Momentan handelt es sich um eine relativ einfache Oracle-SQL*Formsanwendung, die an bestimmten Stellen vom GTDS aufgerufen wird. Auf Wunsch des Anwenders zeigt sie einen Erklärungstext zur aktuellen Meldung an und erlaubt dem Benutzer, einen Kommentar einzugeben. Derzeitig existieren insgesamt zwei Versionen für das

zeichenorientierte und das graphische GTDS. Die folgenden Abbildungen zeigen die beiden Benutzerschnittstellen.

Wer das Nest, ist der V... Leipzig V1

File Edit Setup Control Help

Aktion Editieren Block Feld Datensatz abfraGen Hilfe

UNGELESENE MELDUNGEN

Patient janssen, Johann, 10.10.1980 Nr. 8000

eig.Abt. INNERE MEDIZIN, KRANKENHAUS NR.1, Gießen, Nr. 1

2 UNGELESENE MELDUNG/EN AUS DEM EU-MODUL :

NR. 3 Maligne Tumoren an Dickdarms, Trachea, Lunge oder Bronchus und histologisch kein <Karzinoid> sind bei Patienten unter 20 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich. (zu Herrn janssen, Histologie: 8250 Lokalisation : 33 Alter 17)

NR. 6 Maligne Tumoren der Plazenta mit Histologie <Corionkarzinom> sind bei Patienten jünger 15 und älter 45 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich. (zu Frau Janssen, Histologie: 9100 Lokalisation : 58 Alter 51)

Nachricht :

Bewertung :

schon gelesen

Zählung: *2 <HGM><BSB><DBG> <Ersetz>

Abb. 14 : Oberfläche des Arden-Melders in der zeichenorientierten Umgebung

Übersicht über vorhandene Angaben zum ausgewählten Patienten

Patient: Margret Janssen 04.04.1910 ID.: 8013 Eigene Abteilung: INNERE MEDIZIN, KRANKENHAUS NR.1, Gießen

Ungesene Nachrichten

1 UNGELESENE MELDUNG/EN AUS DEM EU-MODUL :

2 UNGELESENE MELDUNG/EN AUS DEM EU-MODUL :

WARUNG Maligne Tumoren der Plazenta mit Histologie <Corionkarzinom> sind bei Patienten jünger 15 und älter 45 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich. (zu Frau Janssen, Histologie: 9100 Lokalisation : 58 Alter 51)

Bewertung: inhaltlich OK Erklärung: gelesen

Bewertung: Erklärung: gelesen

Zurück

Diagnosedaten Verlaufsdaten Konsil Therapiedaten Abschluß zurück

IARC-Prüfung zur Erkennung falscher/unwahrscheinlicher Alter/Lokalisation/Histologie Kombinationen Count: *1

Abb. 15 : Oberfläche des Arden-Melders in der graphischen Umgebung

4.4.5. Technische Darstellung des Gesamtmoduls

Während die Architekturübersicht in Abb. 10 sich auf die Vorstellung der wesentlichen Funktionseinheiten konzentriert, soll in Abbildung 16 die softwaretechnische Seite im Vordergrund stehen. Sie beinhaltet neben den unterschiedlichen Bestandteilen wie SQL*-Formsanwendungen, Dateien, Tabellen, C-Programmen, Datenbanktrigger und Datenbankprozeduren noch die Steuer- und Datenflüsse zwischen diesen. Weiterhin ist ersichtlich, ob die Bestandteilen innerhalb des RDBMS oder auf Betriebssystemebene definiert wurden.

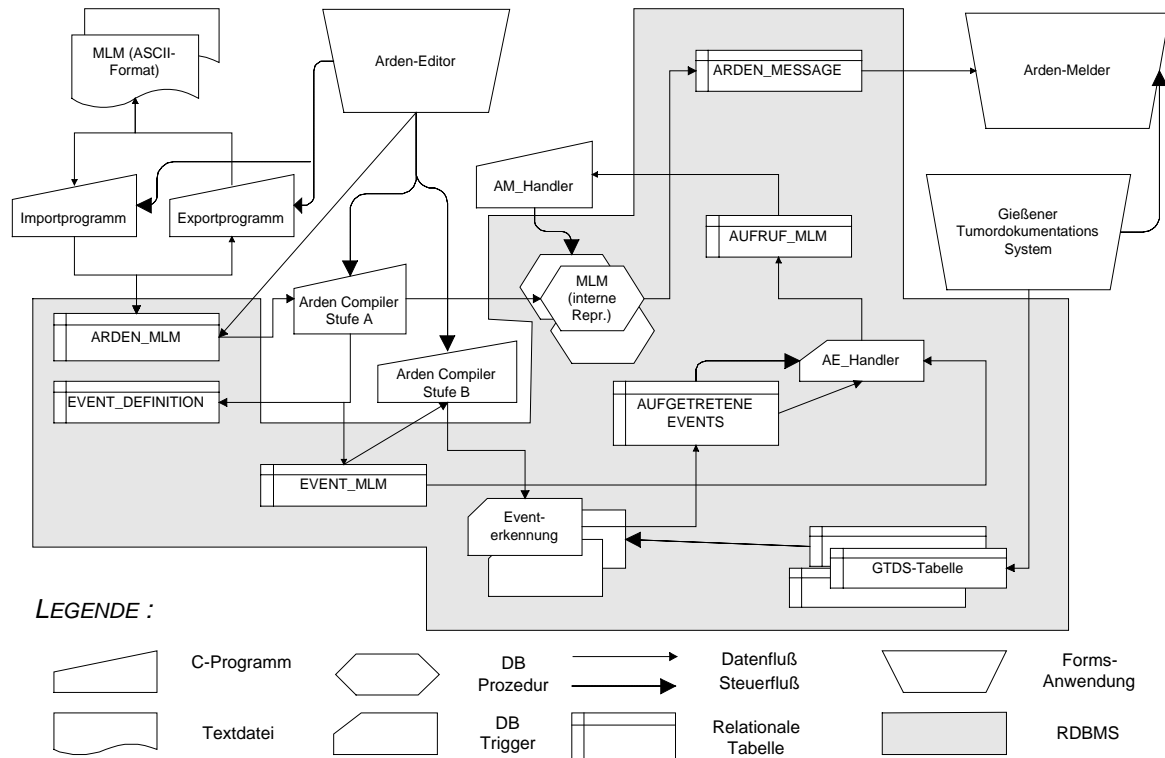


Abb. 16 : Technische Übersicht über das wissensbasierte Modul

In der oberen rechten Ecke der Abbildung sind die Objekte angeordnet, mit denen der GTDS-Benutzer kommuniziert. Es handelt sich um das GTDS selbst sowie um den Arden-Melder zur Präsentation eventueller Meldungen. Die gestrichelte Verbindung zwischen diesen beiden Oracle-Formsanwendungen bedeutet, daß der Arden-Melder vom GTDS aufgerufen wird. Benutzeraktivitäten im GTDS bewirken typischerweise einen Datenfluß zu den entsprechenden Tabellen, die in der Abbildung als Informationssystem-Tabellen bezeichnet sind. Diesen Tabellen zugeordnete Trigger erkennen relevante Ereignisse - relevante Ereignisse sind Ereignisse, die in irgendwelchen MLMs definiert wurden - und tragen deren Auftreten mit zusätzlichen Informationen wie Uhrzeit und die Datenveränderung beschreibende Angaben in die Tabelle `AUFGETRETENE_EVENTS` ein. Die resultierenden Konsequenzen wurden in Kapitel

4.4.3. bei der Darstellung der Arden-Engine erläutert. Resultat der Benutzeraktivität ist schließlich eine eventuelle Meldung, die in der Tabelle `ARDEN_MESSAGE` zunächst als ungelesene Meldung abgelegt wird.

Aus der Abbildung kann entnommen werden,

- daß es sich im wesentlichen um um Objekte des Oracle - RDBMS handelt. Dies erleichtert die Distribution der Arden-Shell unter den Anwendern, die unterschiedlichste Betriebssysteme einsetzen, da die Oracle-Objekte auch über Plattformgrenzen hinweg relativ problemlos portierbar sind und
- daß, obwohl die Shell - logisch gesehen - stark an das GTDS gekoppelt ist, eine physische Abhängigkeit lediglich zu den relationalen Tabellen des GTDS besteht. Veränderungen am GTDS (solange sie nicht die zugrundeliegenden relationalen Tabellen betreffen) haben keine Auswirkung auf die Funktionsweise der Shell. Dies ist insbesondere im Zuge der Umstellung der bisher mit einer zeichenorientierten Oberfläche versehenen GTDS auf eine graphische Oberfläche von großer praktischer Bedeutung.

4.5. Untersuchung der Shell für wissensverarbeitende Funktionen

Die in Kapitel 4.1. definierten Anforderungen an die Shell für wissensverarbeitende Funktionen sind wichtige Bedingungen für die klinische Einsetzbarkeit der mit ihr realisierten wissensverarbeitenden Funktionen. Trotzdem müssen bei der Realisierung bestimmte Regeln eingehalten werden, deren Mißachtung sich negativ auf die klinische Einsetzbarkeit auswirken könnte. Die Herausarbeitung dieser Regeln ist Gegenstand dieses Kapitels.

4.5.1. Die unterschiedlichen Wissenstypen innerhalb eines MLMs

Jedes Medical Logic Module enthält das gesamte Wissen, das zur Durchführung einer konkreten integrierten wissensverarbeitenden Funktion nötig ist. Den dazu nötigen unterschiedlichen Wissenstypen ist gemeinsam, daß sie in den Slots der Knowledge-Categorie lokalisiert sind.

Um den folgenden methodischen Untersuchungen eine begriffliche Basis zur Verfügung zu stellen, soll das in einem Medical Logic Module repräsentierte Wissen gemäß der folgenden Typisierung strukturiert werden :

- *Fachwissen* : hiermit sind Wissenselemente gemeint, die allgemein im Anwendungsgebiet gültig sind. Es handelt sich oft um Fakten, wie beispielsweise „die Niere ist ein paariges Organ“, „die Substanz Cimetidin wird renal ausgeschieden“ oder „die Lokalisation Lymphknoten ist mit einer Histologie aus der Familie der Epithelialtumoren eigentlich immer unverträglich“. Aufgrund seines allgemeingültigen Charakters ist eine Wiederverwendung (Knowledge Reusing) solcher Wissenselemente in anderen Problemkontexten möglich und sinnvoll. Das Fachwissen ist normalerweise implizit im prozeduralen Code des Logic-Slots enthalten.
- *Problemlösungswissen* : hiermit sind Wissenselemente gemeint, die spezifisch für eine konkrete Problemlösung sind und dabei auf das obige Fachwissen zurückgreifen. Es kann in die folgenden Untertypen weiter zerlegt werden.
 - *Verarbeitungswissen*: Dieses Wissenselement beschreibt die Anwendung des obigen Fachwissens auf die Fakten des Informationssystems. Im typischen Fall generiert der hiermit fixierte Ableitungsalgorithmus bei Verstoß der Fakten gegen das Fachwissen eine für den Benutzer bestimmte Meldung. Der Algorithmus kann die Form einer

einfachen Regel, aber auch eines komplexeren Entscheidungsbaumes haben. Das Verarbeitungswissen spiegelt sich im prozeduralen Code des Logic-Slots wieder.

- *Auslösungswissen* : hiermit ist das Wissen über das das MLM auslösende Ereignis gemeint. Durch ihr Vorhandensein wird die Ablaufintegration einer wissensverarbeitenden Funktion ins GTDS möglich. Es ist in der Ereignisdefinition des Data-Slots und im Event-Slot enthalten. Einige der noch folgenden Untersuchungen zeigen den Effekt unterschiedlicher Definitionsmöglichkeiten auf den Logic-Slot und die Systemperformance.
- *Datenintegrationswissen* : hiermit ist das Wissen gemeint, das beschreibt, welche Daten für das Funktionieren des MLMs wichtig sind, und wo diese Daten innerhalb des Informationssystems gefunden werden können. Dieses Wissen ist Voraussetzung für die Integration der wissensverarbeitenden Funktionen in das GTDS. Das Integrationswissen ist im Data-Slot lokalisiert.

Die vorgestellte Typisierung orientiert sich an der von Van der Lei gemachten Zweiteilung in „medical knowledge“ und „critiquing knowledge“ [Van der Lei 1990], wobei das „critiquing knowledge“ noch weiter differenziert wird. Unter ersterem versteht er eine deklarative Beschreibung von Beziehung zwischen Objekten, von denen in beliebigen MLMs die Rede ist. Typisch für dieses "medical knowledge" ist, daß es sich um passive Muster handelt, die erst durch Verwendung durch aktive Prozesse aktiviert werden. Das "critiquing knowledge" beschreibt diese aktiven den Ableitungsvorgang durchführenden Prozesse. Andere Autoren machen analoge Einteilungen. [Prokosch 1993] spricht von "deklarative knowledge" und "procedural knowledge". In seinem vorgestellten Ansatz modelliert er das "deklarative knowledge" innerhalb eines semantischen Netzwerks. [Hripcsak 1995] spricht von den Begriffspaaren "domain knowledge" und "problem solving knowledge". Unter "domain knowledge" versteht er medizinische Fakten, unter "problem solving knowledge" die Fähigkeit durch Anwendung der medizinischen Daten, ein Problem zu lösen.

4.5.2. Auswirkung der Wahl des Eventtyps auf den Logic-Slot

Eine geschickte Wahl des Eventtypes in der Eventdefinition eines MLMs kann sich positiv auf den Inhalt des Logic-Slots auswirken. Dies soll an einem Beispiel verdeutlicht werden. Es geht

um ein MLM, das die Aufgabe hat, die zeitliche Beziehung zwischen Diagnosedatum und Operationsdatum zu überprüfen. Das MLM beinhaltet dazu

- das Fachwissen, daß ein Operationsdatum im Normalfall zeitlich nicht vor dem zugehörigen Diagnosedatum liegt,
- das Problemlösungswissen,
 - wann das MLM abgearbeitet werden soll, nämlich nach Abspeicherung eines Operationsdatums (Auslösungswissen),
 - wo es die benötigten Informationen im Informationssystem finden kann (Datenintegrationswissen),
 - das Verarbeitungswissen, daß bei Nichtübereinstimmung der gelesenen Daten mit dem Fachwissen eine Meldung erfolgen soll.

Im Zuge der Realisierung dieses MLMs wurden zwei unterschiedliche Versionen erstellt, die auf der folgenden Seite in Abbildung 17 vergleichend dargestellt sind. Die sich unterscheidenden Bestandteile sind durch einen kursiven Schrifttyp hervorgehoben. In der ersten Version werden die relativ unspezifischen Ereignisse *INSERT_OP* und *UPDATE_OP* verwendet, die jedesmal nach Anlegen bzw. Modifikation eines Operationssatzes vorliegen. Das MLM der ersten Version, wird also auch dann ausgelöst, wenn kein OP-Datum vorliegt. Dies führt zur Sicherstellung des Vorhandenseins eines OP-Datums innerhalb des Verarbeitungswissens des Logic-Slots. Da die zweite Version des MLMs durch die spezifischeren Ereignisse *UPDATE_OP_DATUM* und *INSERT_OP_DATUM* ausgelöst werden, die nur vorliegen, wenn auch tatsächlich ein OP-Datum existiert, kann hier die Kontrolle auf Vorhandensein eines OP-datums innerhalb des Logic-Slots entfallen

An diesem Beispiel wird deutlich, daß zur Verkürzung und Vereinfachung des Logic-Slots möglichst spezifische Ereignisse für die Auslösung verwendet werden sollten. Dies wirkt sich positiv sowohl auf

- die Nachvollziehbarkeit der Systemaktivität für Benutzer und Wissensverwalter als auch
 - die Erleichterung des Knowledge Sharings
- aus.

Hieraus können folgende Forderungen abgeleitet werden :

- daß bei der *Konzeption* einer Shell für wissensverarbeitende Funktionen basierend auf der Arden-Syntax die Definitionsmöglichkeit unterschiedlich spezifischer Ereignisse wichtig ist, und
- daß beim *Schreiben* von neuen MLMs möglichst spezifische Ereignisse verwendet werden sollten.

VERSION 1	VERSION 2
<p>DATA:</p> <p><i>INSERT_OP</i> := EVENT {insert,operation};</p> <p><i>UPDATE_OP</i> := EVENT {update,operation};</p> <p><i>OP_DATUM</i> := READ last{SQL: select op_datum from operation where fk_tumorfk_patient = test4.fk_tumorfk_patient and fk_tumortumor_id = test4.fk_tumortumor_id and op_nummer = test4.op_nummer};</p> <p><i>DIAGNOSEDATUM</i> := READ last{SQL: select diagnosedatum from tumor where fk_patientpat_id = test4.fk_tumorfk_patient and tumor_id = test4.fk_tumortumor_id};</p> <p>::</p> <p>EVOKE:</p> <p><i>INSERT_OP OR UPDATE_OP</i>;</p> <p>::</p> <p>LOGIC:</p> <p><i>alles_fehlt</i> := false;</p> <p><i>diagnosedatum_fehlt</i> := false;</p> <p><i>opdatum_fehlt</i> := false;</p> <p><i>datums_da</i> := false;</p> <p>if (op_datum is null) and (diagnosedatum is null) then</p> <p><i>alles_fehlt</i> := true;</p> <p>conclude true;</p>	<p>DATA:</p> <p><i>INSERT_OP_DATUM</i> := EVENT {insert,operation,op_datum};</p> <p><i>UPDATE_OP_DATUM</i> := EVENT {update,operation,op_datum};</p> <p><i>OP_DATUM</i> := READ last{SQL: select op_datum from operation where fk_tumorfk_patient = test4.fk_tumorfk_patient and fk_tumortumor_id = test4.fk_tumortumor_id and op_nummer = test4.op_nummer};</p> <p><i>DIAGNOSEDATUM</i> := READ last{SQL: select diagnosedatum from tumor where fk_patientpat_id = test4.fk_tumorfk_patient and tumor_id = test4.fk_tumortumor_id};</p> <p>::</p> <p>EVOKE:</p> <p><i>INSERT_OP_DATUM OR UPDATE_OP_DATUM</i>;</p> <p>::</p> <p>LOGIC:</p> <p><i>diagnosedatum_fehlt</i> := false;</p> <p>;</p>

<pre> else if (op_datum is null) then opdatum_fehlt := true; conclude true; else if (diagnosedatum is null) then diagnosedatum_fehlt := true; conclude true; else if (diagnosedatum >= op_datum) then datums_da:= true; conclude true; else conclude false; endif; endif; endif; endif; ;; ACTION: if alles_fehlt then write "Es fehlen Diagnosedatum und Operationsdatum !"; else if diagnosedatum_fehlt then write "Es fehlt noch das Diagnosedatum !"; else if opdatum_fehlt then write "Es fehlt noch das Operationsdatum !"; else if datums_da then WRITE "Ist es möglich, daß die Operation vor Diagnosestellung durchgeführt wurde ?"; endif; endif; endif; endif; ;; </pre>	<pre> if (diagnosedatum is null) then diagnosedatum_fehlt := true; conclude true else if (diagnosedatum > op_datum) then conclude true; else conclude false; endif; endif; ;; ACTION: if diagnosedatum_fehlt then write "Es fehlt noch das Diagnosedatum zu diesem Fall !"; else WRITE "Ist es korrekt, daß die Operation vor der Diagnosestellung durchgeführt wurde ? "; endif; ;; </pre>
--	---

Abb.17: Versionen eines MLMs zur Illustration der Auswirkung des Eventtyps auf die Länge eines MLMs

4.5.3. Auswirkung der Wahl des Eventtyps auf die Systemperformance

Eine geschickte Wahl des Eventtypes in der Eventdefinition eines MLMs kann sich auch positiv auf die Performance des Systems auswirken. Der Effekt ist leicht zu erklären; er ist darauf zurückzuführen, daß die Arden-Engine bei relativ unspezifischen Events häufiger aktiv wird und das entsprechende MLM auslöst, ohne daß dieses eine Meldung produziert. Dieser Sachverhalt soll am Beispiel einer realisierten Integritätsprüfung dargestellt werden. Diese Integritätsprüfung soll dem Benutzer die Einhaltung der Konvention, daß es zu einem Tumor nur eine als Haupthistologie ausgezeichnete Histologie geben kann, nahelegen.

Es benötigt dazu

- das Fachwissen, daß es zu einem Tumor lediglich eine Haupthistologie geben sollte
- das Problemlösungswissen
 - wann das MLM abgearbeitet werden soll, nämlich unmittelbar nach Dokumentation einer Haupthistologie (Auslösungswissen)
 - wo es die benötigten Informationen im Informationssystem finden kann (Datenintegrationswissen)
 - das Verarbeitungswissen, daß bei Verstoß der gelesenen Werte gegen das Fachwissen eine Meldung erfolgen soll.

Abbildung 18 enthält eine vergleichende Darstellung zweier unterschiedlicher Versionen dieses MLMs.

VERSION 1	VERSION 2
<pre>DATA: INSERT_HISTOLOGIE := EVENT {insert,histologie}; UPDATE_HISTOLOGIE := EVENT {update,histologie}; ANZ_HAUPT := READ count{SQL: select fk_tumorfk_patient from histologie where fk_tumorfk_patient = test6.fk_tumorfk_patient and fk_tumortumor_id = test6.fk_tumortumor_id and haupt_neben = 'H'}; ;; EVOKE: INSERT_HISTOLOGIE OR UPDATE_HISTOLOGIE; ;;</pre>	<pre>DATA: INSERT_HISTOLOGIE_HAUPT := EVENT {insert,histologie,haupt_neben, "H"}; UPDATE_HISTOLOGIE_HAUPT := EVENT {update,histologie,haupt_neben, "H"}; ANZ_HAUPT := READ count{SQL: select fk_tumorfk_patient from histologie where fk_tumorfk_patient = test6.fk_tumorfk_patient and fk_tumortumor_id = test6.fk_tumortumor_id and haupt_neben = „H“}; ;; EVOKE: INSERT_HISTOLOGIE_HAUPT OR UPDATE_HISTOLOGIE_HAUPT; ;;</pre>

LOGIC: IF ANZ_HAUPT > 1 THEN CONCLUDE TRUE; ENDIF; ;; ACTION: WRITE "Nach Konvention darf es nur eine Haupthistologie geben !"; ;;	LOGIC: IF ANZ_HAUPT > 1 THEN CONCLUDE TRUE; ENDIF; ;; ACTION: WRITE "Nach Konvention darf es nur eine Haupthistologie geben (Plausibilitätsprüfung 6) !"; ;;
--	--

Abb. 18: Versionen eines MLMs zur Illustration der Auswirkung des Eventtyps auf die Systemperformance

Das MLM der ersten Version wird durch die Ereignisse `INSERT_HISTOLOGIE` und `UPDATE_HISTOLOGIE` der Typen 1 und 2 ausgelöst. Die Arden-Engine wird dabei veranlaßt, bei jeder beliebigen Datenänderung innerhalb der Histologie-Tabelle das MLM auszulösen. Wird beispielsweise der Differenzierungsgrad einer schon dokumentierte Haupthistologie nachträglich verändert, so liegt ein auslösendes Ereignis vom Typ 2 vor. Das MLM der zweiten Version hingegen wird durch zwei Ereignisse der Typen 5 und 6 ausgelöst. Der Auslösemechanismus ruft lediglich bei Eintrag einer Haupthistologie bzw. bei Veränderung einer Nebenhistologie zu einer Haupthistologie das MLM auf. Das entsprechende MLM wird also ausschließlich in den Fällen abgearbeitet, in denen durch das Hinzufügen einer Haupthistologie eine Verletzung der obengenannten Konvention droht. Bei gleichem Meldungsverhalten führt das zweite MLM hierdurch zu einer geringeren Hintergrundaktivität als das erste MLM.

Während beim Beispiel des vorherigen Kapitels die Bedeutung der Wahl des auslösenden Ereignisses auf die Einfachheit eines MLMs im Vordergrund stand, ging es in diesem Beispiel um eine Reduzierung der Systembelastung.

4.5.4. Knowledge Sharing versus Rückgriff auf separat abgelegtes Fachwissen

Ein definiertes Designziel der Arden-Syntax ist die Fähigkeit zum Austausch von Wissensmodulen zwischen unterschiedlichen Institutionen (Knowledge Sharing). Hieraus folgt die Forderung, daß alle für eine Problemlösung benötigten Wissensbestandteile in einem MLM enthalten sein müssen. In diesem Kapitel soll gezeigt werden, daß der Rückgriff auf außerhalb eines MLMs abgelegtes Fachwissen aus einem MLM heraus

- ohne Modifikationen des Sprachumfanges möglich und
- bei umfangreichem Fachwissen sinnvoll sein kann.

Dies soll anhand zweier MLMs zur Sicherstellung der Plausibilität von Lokalisations- und Histologieangaben nach Dokumentation einer Histologie verdeutlicht werden, die in Abbildung 20 dargestellt sind. Besonderes Augenmerk gilt dem Fachwissen über unverträgliche Histologie-Lokalisationspaare. Im ersten MLM (Version 1) ist dieses Fachwissen (lediglich für die Lokalisation „weibliche Brust“) im prozeduralen Code des Logic-Slot des MLMs enthalten, während bei der zweiten Version (Version 2) das Fachwissen über plausible Paare zentral in einer relationalen Tabelle des zugrundeliegenden Informationssystems abgelegt wurde und über eine im Data-Slot definierte Variable vom Typ Liste dem MLM zur weiteren Verarbeitung zur Verfügung gestellt wurde. Die Abbildung 19 zeigt einen Ausschnitt aus dieser Tabelle. Beim genannten Anwendungsfall ist diese der eigentlichen Philosophie der Arden-Syntax widersprechende Repräsentation medizinischen Wissens sinnvoll, da sie - aufgrund des Umfanges des Wissen - (1) übersichtlicher, (2) einfacher zu warten ist und zudem (3) auch in anderen Problemkontexten (Knowledge Reusing) verwendet werden kann. Andere Problemkontexte wären beispielsweise die Generierung einer an eine Lokalisation angepaßten Histologieliste oder die Durchführung von Datenqualitätskontrollen beim Export von Patientendaten.

<i>LOK</i>	<i>LOK-AUF</i>	<i>HIST</i>	<i>HIST-AUF</i>
143	3	80823	2
143	3	81313	2
143	3	81433	2
143	3	81443	2
143	3	81453	2
143	3	82033	2
143	3	82103	2
143	3	81213	2
143	3	82213	2
143	3	82803	2

Abb. 19: Ausschnitt des separaten abgelegten Fachwissens in einer relationalen Tabelle

Der Widerspruch zur Philosophie der Arden-Syntax besteht darin, daß nun beim Austausch des MLMs der Version 2 der alleinige Austausch des entsprechenden MLMs nicht mehr ausreicht, sondern um das separat abgelegte Fachwissen ergänzt werden muß.

VERSION 1	VERSION 2
<p>DATA:</p> <p>INSERT_HISTOLOGIE := EVENT {insert,histologie,fk_histologie_shis}; UPDATE_HISTOLOGIE := EVENT {update,histologie,fk_histologie_shis}; HIST_CODE := READ last{SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = test8.fk_tumorfk_patient and fk_tumortumor_id = test8.fk_tumortumor_id and lfdnr = test8.lfdnr}; LOK_CODE := READ {SQL: select fk_lokalisierunglok from lokalisierung where fk_tumorfk_patient = test8.fk_tumorfk_patient and fk_tumortumor_id = test8.fk_tumortumor_id}; PAT_NAME := READ last{SQL: select name from patient where pat_id = test8.fk_tumorfk_patient}; PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = test8.fk_tumorfk_patient}; ;;</p> <p>EVOKE:</p> <p>INSERT_HISTOLOGIE OR UPDATE_HISTOLOGIE; ;;</p> <p>LOGIC:</p> <p>if ((lok_code is NULL) or (hist_code is null)) then conclude false; endif; if (pat_geschl = "M") then anrede := "Herrn"; else anrede := "Frau"; endif; if (substr(lok_code,1,3) = "174") then if ((hist_code = "80823") or (hist_code = "81313") or (hist_code = "81433") or (hist_code = "81443") or (hist_code = "81453") or (hist_code = "82033") or (hist_code = "82103") or (hist_code = "82123") or (hist_code = "82213") or (hist_code = "82803") or (hist_code = "82813") or (hist_code = "85293") or (hist_code = "85613") or (hist_code = "81423")) then conclude true; endif; endif; ;;</p> <p>ACTION:</p> <p>WRITE "Bei " anrede " " pat_name " ist die Histologie (Schlüssel: " hist_code ") mit der Lokalisation (Schlüssel: " lok_code ") unvereinbar nach Dr. Wächter (Plausibilitätsprüfung 8) !"; ;;</p>	<p>DATA:</p> <p>INSERT_HISTOLOGIE := EVENT {insert,histologie,fk_histologie_shis}; UPDATE_HISTOLOGIE := EVENT {update,histologie,fk_histologie_shis}; HIST_CODE := READ last{SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = test81.fk_tumorfk_patient and fk_tumortumor_id = test81.fk_tumortumor_id and lfdnr = test81.lfdnr}; LOK_CODE := READ last{SQL: select substr(fk_lokalisierunglok,1,3) from lokalisierung where fk_tumorfk_patient = test81.fk_tumorfk_patient and fk_tumortumor_id = test81.fk_tumortumor_id}; PAT_NAME := READ last{SQL: select name from patient where pat_id = test81.fk_tumorfk_patient}; PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = test81.fk_tumorfk_patient}; HIST_LISTE := READ {SQL: select hist from lo_hi_unvertraeglich where lok = lok_code}; ;;</p> <p>EVOKE:</p> <p>INSERT_HISTOLOGIE OR UPDATE_HISTOLOGIE; ;;</p> <p>LOGIC:</p> <p>if ((lok_code is NULL) or (hist_code is null)) then conclude false; endif; if (pat_geschl = "M") then anrede := "Herrn"; else anrede := "Frau"; endif; if (hist_code in hist_liste) then meldungs_modus := "FEHLER"; conclude true; endif; ;;</p> <p>ACTION:</p> <p>WRITE "Bei " anrede " " pat_name " ist die Histologie (Schlüssel: " hist_code ") mit der Lokalisation (Schlüssel: " lok_code ") definitiv unvereinbar nach Dr. Wächter (Plausibilitätsprüfung 81) !"; ;;</p>

Abb. 20: Versionen eines MLMs zur Illustration der Problematik „Knowledge Sharing versus Rückgriff auf separat abgelegtes Fachwissen“

5. Anwendung der Shell für wissensverarbeitende Funktionen

5.1. Anforderungen an zu realisierende wissensverarbeitende Funktionen

Das entwickelte System erlaubt als Shell den Aufbau einer modularen Wissensbasis aus MLMs und stellt ihre Operationalität sicher. Es bietet also die Möglichkeit der Realisierung wissensverarbeitender Funktionen, wobei es als Shell nicht auf ein bestimmtes Anwendungsgebiet festgelegt ist.

Wissensverarbeitende Funktionen eines Anwendungsgebietes können jedoch nur dann sinnvoll mit der Arden-Syntax basierten Shell realisiert werden, wenn folgende Anforderungen berücksichtigt werden :

- *alle für die wissensverarbeitende Funktion nötigen Fakten müssen im Informationssystem vorhanden sein.* Die Arden-Syntax sieht im Gegensatz zu vielen anderen wissensbasierten Systemen keine Interaktion mit dem Benutzer während des Ableitungsvorganges vor. Einige Forscher auf dem Gebiet [Gao 1987] der Arden-Syntax fordern hier eine Modifikation in den nächsten Versionen des Standards.
- *der Ableitungsvorgang im Logic-Slot sollte relativ leicht nachvollziehbar sein.* Während kurze Ableitungsvorgänge in prozeduraler Form sehr leicht (und allgemein) nachvollziehbar sind, ist das Verständnis langer und/oder die Unsicherheit von Fakten und Regeln berücksichtigender Ableitungsvorgänge kein triviales Problem mehr. Aus Gründen der Nachvollziehbarkeit und Anpassungsfähigkeit durch den Anwender sollte auf solche Ableitungen verzichtet werden. Zur Verbesserung der Nachvollziehbarkeit sollten weiterhin, wie in Kapitel 4.5. dargestellt, möglichst spezifische Ereignisse zur Auslösung verwendet werden.
- *eine hohe Ablaufintegration der wissensverarbeitenden Funktion sollte möglich sein.* Hiermit ist gemeint, daß die Auslösung der wissensverarbeitenden Funktion automatisch aus den Dokumentationsaktivitäten des Benutzers folgt (und nicht explizit angestoßen werden muß). Diese Anforderung wird in der Regel durch wissensverarbeitende Funktionen des Typs Entscheidungsmonitoring erfüllt, da hier die dokumentierte Entscheidung als Auslöser der Problemlösung fungieren kann. Bei wissensverarbeitenden Funktionen des Typs

Entscheidungsunterstützung kann die Wahl eines auslösenden Ereignisses problematisch sein. Lassen sich keine vernünftigen auslösenden Ereignisse finden, so kann ein expliziter Aufruf der wissensverarbeitenden Funktionen aus dem Informationssystem bspw. über einen zu definierenden Knopf in Betracht gezogen werden. Eine weitere Alternative besteht im von den Dokumentationsaktivitäten abgekoppelten Aufruf der Funktionen aus einer eigenen Umgebung heraus. Bei den in Kapitel 5.3. für das Anwendungsgebiet Sicherung der Datenqualität vorgestellten MLMs handelt es sich um typische Funktionen zum Entscheidungsmonitoring, die unmittelbar durch Dokumentationsaktivitäten ausgelöst werden, während das in Kapitel 5.4. vorgestellte MLM für das Anwendungsgebiet Studienmanagement als Prototyp einer Funktion vom Typ Entscheidungsunterstützung explizit über einen zur Auslösung definierten Knopf innerhalb des GTDS bzw. über eine dafür vorgesehene Benutzeroberfläche ausgelöst wird (vgl. Abb. 26 und 27). Eine derartige Anwendungsstrategie wissensverarbeitender Funktionen nutzt zwar nicht die Fähigkeit der Arden-Syntax, ablaufintegrierte Funktionen zu ermöglichen, findet seine Berechtigung jedoch in der vorhandenen Nachfrage nach solchen Unterstützungen.

- *die wissensverarbeitende Funktion muß auf die zeitliche Nähe der Dokumentation zur Patientenversorgung abgestimmt sein.* Je nach zeitlicher Nähe können unterschiedliche Funktionen sinnvoll sein. Bei Arztarbeitsplatzsystemen als Prototyp der behandlungsbegleitenden Dokumentation können beispielsweise Funktionen zur Unterstützung der Behandlung durch Leitlinien verwendet werden, während bei einer zeitverzögerten Dokumentation in einem Tumorzentrum der Behandlungsprozeß schon abgeschlossen ist und keiner Unterstützung bedarf. Eine Berücksichtigung dieses Aspektes ist gerade beim GTDS als zugrundeliegendem Informationssystem von Bedeutung, da je nach Anwender die zeitliche Nähe der Dokumentation zur Versorgung variieren kann.

5.2. Mögliche Anwendungsgebiete der Arden-Shell innerhalb des GTDS

Mögliche Anwendungsgebiete der Arden-Shell sind

1. *Erfassungsnahe Sicherung der Datenqualität durch Integritätsprüfungen.* Dieses Anwendungsgebiet wurde im Rahmen dieser Arbeit schwerpunktmäßig behandelt. Jede Integritätsprüfung kann als eine wissensverarbeitende Funktion realisiert werden, zu der sämtliche relevanten Fakten aus dem Informationssystem gelesen werden können. Sie kann als typische Funktion vom Typ Entscheidungsmonitoring problemlos in den Ablauf integriert werden. Besonders hervorzuheben ist, daß ihre Verwendung unabhängig von der zeitlichen Nähe der Dokumentation zur Behandlung sinnvoll ist. In Kapitel 5.3. wird dieses Anwendungsgebiet vertieft und realisierte Integritätsprüfungen diskutiert.
2. *Hinweis auf Alarmsituationen.* Funktionen zu diesem Anwendungsgebiet lassen sich gut mit einem auf der Arden-Syntax basierten System realisieren. Dies ist darauf zurückzuführen, daß es sich um typische Funktionen des Typs Entscheidungsmonitoring handelt und in der Regel keine komplexen Ableitungsvorgänge benötigt werden. In [Hripcsak 1994] wird ein MLM dieses Anwendungsbereiches vorgestellt, welches nach jedem neuen Blutbild basierend auf den vorhandenen Hämatokrit-Werten entscheidet, ob der aktuelle Hämatokrit-Wert schnell fällt oder absolut gesehen unter einem kritischen Wert liegt. Im positiven Fall wird die Meldung „The patient's hematocrit is low or falling rapidly“ ausgegeben.
3. *Zuweisung von Patienten zu Studien.* Die Funktionen dieses Anwendungsgebiet gehören zum Typ Entscheidungsunterstützung. Da auf Seiten der GTDS-Anwender ein starker Bedarf an Unterstützung beim Management onkologischer Studien existiert, wird in Kapitel 5.4. ein für diese Funktion realisiertes MLM hinsichtlich Algorithmus und Anwendungsstrategie diskutiert.
4. *Hinweis auf Behandlungsleitlinien.* Die Funktionen dieses Anwendungsgebietes gehören wie auch die des vorherigen Anwendungsgebietes zum Typ Entscheidungsunterstützung. Voraussetzung für den sinnvollen Entwurf entsprechender Funktion ist eine behandlungsbegleitende Dokumentation im Rahmen eines Arztarbeitsplatzes sowie das Vorhandensein anerkannter strukturierter Leitlinien. Auf eine vertiefende Darstellung wurde im Rahmen dieser Arbeit verzichtet.

5.3. Anwendungsfeld: Sicherung der Datenqualität

5.3.1. Einleitung

Die in einem Klinischen Register anfallenden Daten finden Verwendung bei der Unterstützung der unmittelbaren Patientenversorgung sowie stellen das Ausgangsmaterial für Forschungsvorhaben und Qualitätssicherungsmaßnahmen zur Sicherung von Prozeß- und Ereignisqualität im Gesundheitswesen dar. Eine verlässliche Datenqualität dieses multiple verwendeten Datenbestandes ist eine wichtige Grundlage qualitätssichernder Tumordokumentation [Dudeck 1996].

Es können unterschiedliche Aspekte der Datenqualität unterschieden werden :

- *inhaltliche Aspekte:* Hier geht es darum, ob das Bild, welches die Daten über einen Sachverhalt widerspiegeln, auch tatsächlich der Realität entspricht. Dieser Aspekt wird oft auch als Widerspruchsfreiheit, Plausibilität oder Validität [Parkin 1994] der Daten bezeichnet.
- *Vollständigkeits-Aspekte:* Bei diesem Aspekt steht die Vollständigkeit der Dokumentation im Vordergrund. Im Idealfall sollten alle vorgesehenen Merkmale auch erfaßt worden sein. Entsprechende Fehler liegen vor, wenn bei Vorhandensein bestimmter Merkmale zwingend mit diesen einhergehende Dokumentationsbestandteile fehlen. Bei diesen Dokumentationsbestandteilen kann es sich um Einzelmerkmale (innerhalb eines Diagnosedokumentes sollte ein Diagnosedatum vorhanden sein), aber auch um ganze Merkmalsgruppen (Bsp. bei TNM-Stadium M=1 sollte eine detaillierte Metastasebeschreibung vorliegen) handeln.
- *formale Aspekte:* Hier werden folgende Aspekte subsumiert :
 - Berücksichtigung grundlegender Anforderungen des verwendeten Datenmodells, wie beispielsweise der Eindeutigkeit des Primärschlüssels einer Tabelle (Schlüsselintegrität) beim relationalen Datenmodell,
 - Verwendung gültiger Codes bei Merkmalen, denen ein Schlüsselssystem zugrundeliegt.

Die Shell für wissensverarbeitende Funktionen wurde zur Sicherstellung der inhaltlichen und der die Vollständigkeit betreffenden Aspekte verwendet. Prinzipiell kann die Verwendung der Shell auch auf den Bereich der formalen Aspekte ausgedehnt werden. Dies ist jedoch nicht sinnvoll, da es im Umfeld von Datenbanken und Datenbankapplikationen schon bewährte Methoden zur Sicherstellung formaler Aspekte der Datenqualität gibt. Als Beispiele sollen hier die sogenannten deklarativen Integritätsbedingungen auf Datenbankebene oder die auch die Dateneingabe erleichternden Wertelisten auf Applikationsebene genannt werden.

5.3.2. Semantische Integritätsprüfungen als Methode zur Sicherstellung der Datenqualität

Eine Methode zur Sicherstellung von Datenqualität ist die Durchführung von semantischen Integritätsprüfungen. So verwenden nach [Hilsenbeck 1990] 95% der nordamerikanischen Tumorregister in irgendeiner Form semantische Integritätsprüfungen („edit checks“). Bei semantischen Integritätsprüfungen handelt es sich um eine auf bestimmte Daten angewendete Prozedur zur Erkennung der Erfüllung oder Verletzung von formulierten Integritätsbedingungen. [Haag 1992] definiert diese folgendermaßen :

"Eine Semantische Integritätsbedingung ist ein Sachverhalt, der entweder wahr (erfüllt) oder falsch (verletzt) ist. Sie lassen sich für einzelne Merkmale oder für einzelne Datenstrukturen angeben, können sich aber auch auf zwischen Datenstrukturen bestehende Sachverhalte beziehen."

Eine Integritätsprüfung kann als ein Test auf Vorhandensein eines Fehlers bei gegebenen Daten aufgefaßt werden. Die möglichen sich bei ihrer Anwendung ergebenden Fälle können -analog zu diagnostischen Tests oder Retrievalsystemen - in einer Vierfeldertafel übersichtlich dargestellt werden.

	Dokumentation fehlerhaft	Dokumentation korrekt
Test positiv (Annahme falscher Daten)	Fall A (Fehlermeldung bei fehlerhafter Dokumentation)	Fall B (Fehlermeldung bei korrekter Dokumentation)
Test negativ (Annahme korrekter Daten)	Fall C (keine Fehlermeldung bei fehlerhafter Dokumentation)	Fall D (keine Fehlermeldung bei korrekter Dokumentation)

Der Vorteil der Methode liegt darin, daß sie mit Unterstützung der elektronischen Datenverarbeitung vor Ort ohne großen Arbeitsaufwand durchgeführt werden kann. Bei ihrer Verwendung dürfen jedoch folgende methodische Einschränkungen nicht übersehen werden :

1. Es ist möglich, daß fehlerhafte Daten nicht als solche erkannt werden (obiger Fall C).

Hierfür können zwei Gründe genannt werden. Zum einen, weil es Dokumentations-

bestandteile gibt, für die aufgrund fehlender Integritätsbedingungen keine Integritätsprüfungen formuliert werden können, zum anderen weil Integritätsprüfungen logisch korrekte nicht von im Einzelfall korrekten Daten unterscheiden können. Als Beispiel hierfür können Integritätsprüfungen zur Konsistenz von Histologieangaben zu schon dokumentierten Tumorlokalisationen angeführt werden. Eine im konkreten Fall falsche Histologie bei einem Lippentumor kann nicht als solche erkannt werden, da sie zufällig einer der 58 möglichen Histologien [Van Holten 1991] entspricht. In [Skeet 1991] wird dieser Sachverhalt mit folgenden Worten beschrieben „*While some errors can be detected using range and consistency checks, others cannot because, though actually incorrect, the item may appear quite satisfactory*“

2. Es ist möglich, daß korrekte Daten als fehlerhafte eingestuft werden (Fall B). Dies liegt typischerweise an der *Unsicherheit* und *Unvollständigkeit* des einer Integritätsprüfung zugrundeliegenden Wissens. Zur Unsicherheit nennt Skeet hierzu beispielhaft einige Kombinationen von Patientenalter und Lokalisationen eines vorhandenen Primärtumors [Skeet 1991]. Als Beispiel für zugrundeliegendes unvollständiges Wissen soll die Menge der verträglichen Histologie-Lokalisationspaare genannt werden. Eine Berücksichtigung dieses Umstandes führt zwangsläufig zur Unterscheidung unterschiedlicher Sicherheitsniveaus (beispielsweise in Fehler, Warnungen und Informationen) bei resultierenden Meldungen, um Akzeptanzprobleme zu minimieren.

Bei der EDV-gestützten Durchführung von semantischen Integritätsprüfungen können nach [Skeet 1991] online-basierte von offline-basierten Ansätzen unterschieden werden. Bei ersteren werden die Daten erfassungsnah kontrolliert, während beim offline-basierten Ansatz die Daten zu einem späteren Zeitpunkt einer Überprüfung unterzogen werden. Die Vorteile der erfassungsnahen Kontrolle liegen im oft reduzierten Arbeitsaufwand zur Abklärung und ggf. Entfernung des Fehlers sowie in einer verbesserten Verwendbarkeit der Daten für klinische Zwecke. [Van Holten 1991] bezeichnet eine solche Funktionalität bereitstellende Arbeitsplätze als „*intelligent data entry terminals*“ und attestiert ihnen eine wachsende Popularität.

5.3.3. Realisierung von Integritätsprüfungen mit der Shell für wissensverarbeitende Funktionen

Die entwickelte Shell erlaubt eine erfassungsnahe Durchführung von Integritätsprüfungen unmittelbar während der Eingabe der zu dokumentierenden Daten am GTDS. Es ist also auf ein Anwendungsfeld, in dem die Daten direkt von Dokumentationskräften oder Ärzten eingegeben werden, zugeschnitten.

Die Online-Anwendungsstrategie bedingt eine Festlegung eines oder mehrerer Ereignisse, die zur Durchführung einer Integritätsprüfung führen. Bei dieser Festlegung sind, zur Verbesserung der Benutzerakzeptanz, insbesondere die Berücksichtigung zweier Aspekte wichtig :

- möglichst frühe Meldung einer Integritätsverletzung nach deren Entstehung
- möglichst wenige unpassende Meldungen generieren.

Die Berücksichtigung des zweiten Aspektes hat

- zur Unterscheidung von Meldungen gemäß ihrer Dringlichkeit geführt. Die in der Arden-Syntax vorgesehenen Möglichkeiten der Definition einer PRIORITY oder einer URGENCY sind hier nicht ausreichend, da es sich um MLM-spezifische und statische Angaben handelt (d.h. erstens, daß eine Unterscheidung *unterschiedlicher* von *einem* MLM generierten Meldungen nicht vorgesehen ist und zweitens, daß zur Ausführungszeit eines MLMs diese Angaben nicht verändert werden können). Die Shell wurde so konzipiert, daß eine dynamische Einstufung jeder generierten Meldung in die Kategorien „Information“, „Warnung“ und „Fehler“ möglich ist. Hierdurch wird erreicht, daß
 - der Benutzer auf den ersten Blick die Relevanz einer Meldung erfassen kann, da jede Meldungskategorie durch ein eigenes optisches Element gekennzeichnet ist und
 - die Meldungskomponente ggf. bei bestimmten Benutzergruppen bestimmte Meldungen herausfiltern kann.

Die Abbildung 21 enthält eine Übersicht über alle realisierten und der Verbesserung der Datenqualität dienenden MLMs. Die acht letzten MLMs decken sämtliche vom „International Agency for Research on Cancer“ vorgeschlagenen Prüfungen für Krebsregister ab [Parkin 1994]. Die ersten MLMs berücksichtigen die Belange des relativ umfassenden Datenmodells des Gießener Tumordokumentationssystems. In den folgenden Unterkapiteln dieses Kapitels werden die in den MLMs formulierten Integritätsprüfungen beschrieben. Anhang C enthält den Arden-Syntax-Code der MLMs.

Name	BESCHREIBUNG
ip1	Anzahl der Haupthistologien pro Tumorfall
ip2	zeitliche Beziehung Diagnose/Operationsdatum
ip3	Vollständigkeit der Metastasendokumentation
ip4	Konsistenz der Verlaufshistologie mit vorhandenen Histologien
ip5	Stadiumklassifikation T1c nur bei Ovarial/Mammakarzinomen
ip6	Verwendung der ANN-ARBOR-Stadienklassifikationen nur bei Lymphomen
ip7	Ausschluss von Mehrfachdokumentation eines Tumors
iarc0	zeitliche Beziehung Diagnose/Geburtsdatum
iarc2	Konsistenz von Alter/Tumorlokalisierung/Tumorhistologie
iarc3	Konsistenz von Alter/Tumorlokalisierung
iarc4	Konsistenz von Alter/Tumorhistologie
iarc5	Konsistenz von Geschlecht/Tumorlokalisierung
iarc6	Konsistenz von Geschlecht/Tumorhistologie
iarc10	Konsistenz von Tumorlokalisierung/Tumorhistologie basierend auf möglichen Kombinationen
iarc11	Konsistenz von Tumorlokalisierung/Tumorhistologie basierend auf sehr unwahrscheinlichen Kombinationen

Abb.21: Übersicht über MLMs zur Unterstützung des dokumentierenden Personals bei der Erkennung von Verletzungen der Datenqualität

5.3.3.1. Kontrolle der Anzahl von Haupthistologien

Diese einfache Integritätsprüfung versucht, die in der Tumorbasisdokumentation [Wagner 1983] formulierte Konvention sicherzustellen, nach der innerhalb eines Tumorfalles lediglich eine Histologie als Haupthistologie gekennzeichnet sein darf. Ausgelöst wird sie unmittelbar nach Auftreten einer neuen Haupthistologie im GTDS. Liegen zum entsprechenden Tumorfall mehr als eine Haupthistologie vor, so erfolgt die als Fehler eingestufte Meldung *"Bei `"/anrede/"` `"/pat_name/"` wurde die Dokumentationskonvention, daß es nur eine Haupthistologie pro Tumorfall geben darf, verletzt (Integritätsprüfung 1) !"*.

5.3.3.2. Kontrolle der zeitlichen Sequenz von Diagnose- und Operationsdatum

Diese Integritätsprüfung ist ein exemplarisches Beispiel für die Sicherstellung einer korrekten zeitlichen Folge zweier Datumsangaben am Beispiel des Diagnose- und Operationsdatums. Dem MLM liegt das Wissen zugrunde, daß ein Operationsdatum normalerweise zeitlich auf das Diagnosedatum des entsprechenden Tumors folgt. Die Integritätsprüfung wird nach Dokumentation (Neueingabe, Modifikation) einer Operation ausgelöst. Es werden gegebenenfalls drei unterschiedliche Meldungen generiert:

1. liegt kein Diagnosedatum vor, so erfolgt ein entsprechender Hinweis als Warnung,
2. liegt das Operationsdatum vor dem Diagnosedatum des entsprechenden Tumors, so erfolgt die Meldung *"Bei `"/anrede/"` `"/pat_name/"` wurde eine Operation vor der Diagnosestellung durchgeführt wurde. Bitte kontrollieren ! (Integritätsprüfung 1)"* als Fehler,
3. sind Operationsdatum und Diagnosedatum identisch, so wird ein Hinweis auf den diagnostischen Charakter der durchgeführten Operation als Information generiert.

5.3.3.3. Vollständigkeitskontrolle von Metastasenangaben in TNM-Stadium und Metastasenbeschreibung

Diese Integritätsprüfung versucht die Vollständigkeit der Dokumentation bezüglich Metastasenangaben zu erreichen. Sind bei einem Tumorfall Metastasen aufgetreten, so sollte dies gemäß der Tumorbasisdokumentation zum einen zu einer entsprechenden TNM-Stadienbeschreibung und zum anderen zu einer detaillierten Metastasebeschreibung (einschließlich Lokalisation) führen. Vollständigkeitsfehler dieser Art geschehen bei unerfahrenen Anwendern recht oft, da diese nicht unbedingt wissen, daß inhaltlich ähnliche Angaben noch an einer anderen Stelle dokumentiert werden müssen. Die realisierte Integritätsprüfung wird zeitverzögert alternativ nach Dokumentation (Neueingabe, Modifikation) eines entsprechenden TNM-Stadiums oder nach Dokumentation (Neueingabe, Modifikation) einer detaillierten Metastasenbeschreibung ausgelöst. Durch die Zeitverzögerung wird die Anzahl von unnötigen Meldungen an den Benutzer reduziert, indem etwas abgewartet, bis die zur Vollständigkeit notwendigen Daten möglicherweise nachdokumentiert worden sind. Konkret heißt dies, daß nach Dokumentation eines metastasespezifischen TNM-Stadiums mit der Durchführung der Integritätsprüfung abgewartet wird, bis die detaillierte Metastasebeschreibung - sollte an sie gedacht worden sein - abgeschlossen worden ist, und daß bei umgekehrtem Vorgehen des Anwenders nach Dokumentation einer detaillierten Metastasebeschreibung ein Nachholen der TNM-Beschreibung ermöglicht wird.

Abbildung 22 zeigt die Entscheidungsrichtlinien zur Generation der unterschiedlichen Meldungstexte in Abhängigkeit von den vorhandenen Angaben zu metastasespezifischen TNM-Stadien und detaillierteren Metastasebeschreibungen im System. Die Entscheidungsrichtlinien differenzieren nach tumorspezifischer und tumorunspezifischer Auslösung der Prüfung. Dies ist nötig, da Metastasen im klinischen Alltag nicht immer einem Tumor zugeordnet werden können und das GTDS-Datenmodell diesem Umstand durch eine optionale Zuordnung Rechnung trägt. Da trotz Berücksichtigung einer Verzögerung der Auslösung der Integritätsprüfung eine unpassende Meldung generiert werden kann (das Benutzerverhalten kann von Benutzer zu Benutzer oder gar für einen Benutzer variieren, so daß die vorgesehene Zeitverzögerung möglicherweise nicht ausreicht), wird diese als Warnung eingestuft.

Ausgelöst nach Dokumentation	ANZ_MET_B	ANZ_TNM_MET	Meldung
metastasespezif. TNM-Stadium	gleich Null	größer Null	Meldung 1
s.o.	größer Null	größer Null	keine Meldung
s.o.	größer Null	größer Null	keine Meldung
Tumorspezif. Metastase	größer Null	größer Null	keine Meldung
s.o.	größer Null	gleich Null	Meldung 2
Tumorunspezif. Metastase	größer Null	gleich Null	Meldung 2
s.o.	größer Null	größer Null	keine Meldung

wobei	
ANZ_MET_B:	Anzahl von Metastaselokalisationen (ML); nach tumorspezifischer Auslösung die MLs zum Tumorfall <i>einschließlich</i> aller tumorunspezifischen ML, nach tumorUNspezifischer Auslösung alle MLs zum Patienten
ANZ_TNM_MET :	Anzahl aller auf metastasespezif. TNM-Beschreibungen; nach tumorspezifischer Auslösung zum Tumorfall, nach tumorUNspezifischer Auslösung pauschal zum Patienten
M 1 :	<i>"Bei " anrede " " pat_name " wurde zum Tumor ein TNM-Stadium dokumentiert, das auf eine Metastase hinweist. Vergessen Sie bitte nicht eine detailliertere Metastasenbeschreibung im Block Metastase. (Plausibilitätsprüfung 11)."</i>
M 2 :	<i>"Bei " anrede " " pat_name " wurde zum Tumor eine Metastasenbeschreibung durchgeführt. Eigentlich müßte sich dies in TNM-Angaben widerspiegeln. (Plausibilitätsprüfung 11)."</i>

Abb. 22 : Darstellung der Entscheidungsrichtlinien zur Generation von Meldungen

5.3.3.4. Kontrolle auf Verträglichkeit von Verlaufshistologien mit schon vorhandenen Histologien

Diese Integritätsprüfung versucht sicherzustellen, daß dokumentierte Verlaufshistologien sich von schon dokumentierten Histologien zum Tumorfall lediglich durch ihre Dignität unterscheiden. Ein Abweichen dieser Regel ist meistens auf einen falschen Tumorkontext zurückzuführen. Die Integritätsprüfung wird unmittelbar nach Dokumentation (Neueingabe, Modifikation) einer Histologie ausgelöst. Da in Ausnahmefällen bei einem Tumorfall mehrere nicht nur durch die Dignität sich unterscheidende Histologien möglich sind, wird eine generierte Meldung nicht als Fehler, sondern lediglich als Warnung eingestuft.

5.3.3.5. Verträglichkeitskontrolle von TNM-Stadium T1c mit der Lokalisation des Tumors

Diese Integritätsprüfung demonstriert die Überwachung von TNM-Dokumentationskonventionen. Hierbei wird versucht sicherzustellen, daß die Codierung „T1c“ aus [UICC 1997] lediglich bei Ovarial- und Mamakarzinomen verwendet wird. Die Integritätsprüfung wird unmittelbar durch Ereignisse vom Typ 5 und 6 (den spezifischsten der Shell) ausgelöst, die dann vorliegen, wenn ein neuer TNM-Satz mit „T1c“-Stadiencodierung bzw., wenn in einem vorhandenen TNM-Satz eine „T1c“-Stadiencodierung eingetragen wurde. Bei Verstoß gegen die Konvention erfolgt die als Fehler eingestufte Meldung *"Bei `"/anrede/"` liegt ein Tumor an folgenden Lokalisationen vor: `"/lok_text/"`. Der verwendete TNM-Kode T1c ist nur bei Ovarial/ und Mamakarzinomen vorgesehen. (Integritätsprüfung 5)"*. Vergleichbare Prüfungen können leicht in analoger Art und Weise entwickelt werden.

5.3.3.6. Verträglichkeitskontrolle der Verwendung der ANN-ARBOR-Stadienklassifikation mit Tumorfall

Diese einfache Integritätsprüfung versucht sicherzustellen, daß die ANN-ARBOR-Stadienklassifikation lediglich bei Lymphomen Verwendung findet. Sie wird unmittelbar nach Dokumentation (Neueingabe, Modifikation) einer ANN-ARBOR Stadienklassifikation ausgelöst. Liegen keine Histologieangaben zum Tumorfall vor, so wird die als Warnung eingestufte Meldung *"Beim Patienten `"/pat_name/"` haben Sie eine ANN-ARBOR Stadienklassifikation durchgeführt, ohne daß Histologien vorliegen. Nach Möglichkeit nachholen ! (Integritätsprüfung 6)"* . Bei Vorhandensein einer nicht systemerkrankungsspezifischen Histologie erfolgt die als Fehler eingestufte Meldung *"Sie haben beim Patienten `"/pat_name/"` eine ANN-ARBOR Stadienklassifikation durchgeführt, ohne daß eine lymphomspezifische Histologie (`"/hist_liste/"`) vorliegt. Bitte kontrollieren ! (Integritätsprüfung 6)"* .

5.3.3.7. Ausschluß von Mehrfachdokumentation eines Tumors

Hier geht es um die Sicherstellung, daß es sich bei einem als Zweittumor dokumentierten Tumor auch wirklich um einen neuen Tumor handelt. Gerade in Anbetracht der unterschiedlichen Datenquellen eines krankheitsorientierten Informationssystems ist es unerlässlich zu prüfen, ob es sich bei einem gemeldeten Tumor um einen mehrfach gemeldeten

Ersttumor oder um einen Zweittumor handelt. Um diese Entscheidung auf eine verlässliche Grundlage zu stellen, sollen die Entscheidungsleitlinien der IARC (International Agency For Research On Cancer) verwendet werden [Muir 1991]. Die folgende Abbildung verdeutlicht die Entscheidungsrichtlinien der IARC :

Anz. prim.Tumore	Lokalisation (ICD-0 3-stell.)	Histologie (ICD-O 3-stell.)	Diagnosedatum	Anz. zu registr. Tumore
2 oder mehr	gleich	gleich	gleich	1
2 oder mehr	gleich	gleich	verschieden	1
2 oder mehr	gleich	verschieden	gleich	2 oder mehr
2 oder mehr	gleich	verschieden	verschieden	2 oder mehr
2 oder mehr	verschieden	gleich	gleich	2 oder mehr
2 oder mehr	verschieden	gleich	verschieden	2 oder mehr
2 oder mehr	verschieden	verschieden	gleich	2 oder mehr
2 oder mehr	verschieden	verschieden	verschieden	2 oder mehr

Abb. 23 : Richtlinien zur Erkennung von Mehrfachdokumentationen eines Tumors der IARC

Wie die obere Abbildung zeigt, besteht die Ausgangssituation jeweils darin, daß zwei oder mehrere Tumor für einen Patienten dokumentiert sind. Die Kategorien Lokalisation, Histologie und Diagnosedatum entscheiden je nach vorliegender Kombination darüber, wieviele Tumoren zu dokumentieren sind. Eine weitere hier nicht berücksichtigte Kategorie stellt die Seitenlokalisation bei paarigen Organen dar.

Bevor darauf eingegangen wird, was unter gleichen bzw. verschiedenen Lokalisationen, Histologien oder Diagnosedaten zu verstehen ist, soll darauf hingewiesen werden, daß das oben dargestellte Schema ohne Beeinflussung der Entscheidungsrichtlinie durch Weglassen der Kategorie Diagnosedatum vereinfacht werden kann: stimmen in mindestens einer der Kategorien Lokalisation und Histologie die vorhandenen Tumoren nicht überein, so sind sie als unterschiedliche Tumoren zu dokumentieren.

Gleiche Lokalisation bedeutet Gleichheit auf der dreistelligen Ebene der ICD (z .B. ICD 151.3 und ICD 151.4 aus [WHO 1976]) stimmen auf dreistelliger Ebene überein und würden bei übereinstimmender Histologie als ein Tumor dokumentiert).

Bei Übereinstimmung der ersten drei Stellen des Morphologie-Codes wird von einer gleichen Histologie gesprochen. Bei dieser Vorgehensweise würden z.B. ein verhornendes Plattenepithelkarzinom (8071/3 aus [WHO 1990]) und ein spindelzelliges Plattenepithelkarzinom (8074/3 aus [WHO 1990]) als histologisch gleich gelten.

Das Diagnosedatum wird als gleiches Datum gewertet, wenn die betrachteten Datumsangaben nicht mehr als zwei Monate auseinanderliegen.

Das realisierte MLM prüft bei Eintrag eines Diagnosesatzes, ob zum Patienten andere Tumorfälle vorliegen. Im positiven Fall prüft es auf Gleichheit der jeweiligen Lokalisationen und Histologien gemäß der von der IARC vorgegebenen Definition von Gleichheit und gibt entsprechend der Entscheidungsrichtlinie ggf. einen Hinweis auf eine Mehrfachdokumentation ein und desselben Tumors aus. Falls im aktuellen Tumorfall keine Lokalisation bzw. Histologieangaben dokumentiert sind, so erfolgt ein Hinweis auf Unvollständigkeit der Dokumentation.

5.3.3.8. Kontrolle der zeitlichen Sequenz von Diagnose- und Geburtsdatum

Dem MLM liegt das triviale Wissen zugrunde, daß ein Diagnosedatum normalerweise zeitlich nach dem Geburtsdatum des entsprechenden Patienten liegen muß. Die Integritätsprüfung wird nach Neueingabe eines Diagnosesatzes sowie nach Modifikation eines Diagnosedatums ausgelöst. Es werden gegebenenfalls vier unterschiedliche Meldungen generiert:

1. liegt kein Diagnosedatum vor, so erfolgt ein entsprechender Hinweis als Information,
2. liegt kein Geburtsdatum vor, so erfolgt ein entsprechender Hinweis als Information,
3. liegt das Diagnosedatum vor dem Geburtsdatum, so wird die Meldung *"Bei `"/anrede/"` `"/pat_name/"` liegt das Diagnosedatum zeitlich VOR Geburtsdatum ! (Integritätsprüfung 8)"* als Fehler generiert,
4. liegt das Diagnosedatum nach dem aktuellen Tagesdatum, so wird die Meldung *"Bei `"/anrede/"` `"/pat_name/"` liegt das Diagnosedatum zeitlich NACH dem Tagesdatum ! (Integritätsprüfung 8)"* als Fehler generiert.

Die Integritätsprüfung deckt also neben der Inkonsistenz von Diagnose- und Geburtsdatum noch Unvollständigkeiten in der Dokumentation auf.

5.3.3.9. Verträglichkeit von Alter, Lokalisation und Histologie eines Tumors

Diese Integritätsprüfung berücksichtigt alle IARC-Integritätsbedingungen aus [Parkin 1994], in denen das Patientenalter sowie Histologien und Lokalisationen des Tumors eine Rolle spielen.

Da die definierten Datenkonstellationen lediglich als unwahrscheinlich eingestuft sind, werden sämtliche Meldungen als Warnungen generiert. Die Integritätsprüfung wird nach Neueingabe oder Modifikation von Lokalisationen sowie von Histologien ausgelöst. Je nach Sachverhalt werden im Fehlerfall alternativ folgende (hier etwas verkürzt aufgeführte) Meldungen ausgegeben :

- *Maligne Tumoren der Prostata mit Histologie <Adenokarzinom> sind bei Patienten unter 40 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich*
- *Maligne Tumoren des Dünndarms ohne Histologie <Lymphom> sind bei Patienten unter 20 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich*
- *Maligne Tumoren an Dickdarms, Trachea, Lunge oder Bronchus und histologisch kein <Karzinoid> sind bei Patienten unter 20 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich.*
- *Maligne Tumoren des Auges und Augenanhanges mit Histologie <Retinoblastom (o.n.A. oder undifferenziert>) sind bei Patienten älter als 5 Jahre (Integritätsprüfung IARC) sehr unwahrscheinlich.*
- *Maligne Tumoren der Plazenta mit Histologie <Corionkarzinom> sind bei Patienten jünger als 15 und älter als 45 Jahre (Integritätsprüfung IARC) sehr unwahrscheinlich.*

Alle Meldungen enden mit einer Auflistung des Patientennamen, der Lokalisations- und Histologiedaten sowie des Patientenalters. Das der Prüfung zugrundeliegende Wissen über unwahrscheinliche Datenkonstellationen kann entweder aus [Parkin 1994] oder aus dem prozeduralen Code des entsprechenden MLMs im Anhang C entnommen werden.

5.3.3.10. Verträglichkeit von Patientenalter und Lokalisation eines Tumors

Für diese Integritätsprüfungen gelten die gleichen Bemerkungen, wie für das vorherige MLM, mit dem Unterschied, daß diese Prüfung nicht nach Dokumentation einer Histologie ausgelöst wird. Die gegebenenfalls generierten Meldungen sehen folgendermaßen aus :

- *Maligne Tumoren an der/n Lokalisation/en ("||LOK_LISTE_D_TEXT||") sind bei Patienten unter 5 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich.*
- *Maligne Tumoren an der/n Lokalisation/en ("||LOK_LISTE_D_TEXT||") sind bei Patienten unter 20 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich.*

Die Variable LOK_LISTE_D_TEXT enthält eine Beschreibung der dokumentierten Lokalisation/en.

5.3.3.11. Verträglichkeit von Patientenalter und Histologie eines Tumors

Für diese Integritätsprüfungen gelten die gleichen Bemerkungen, wie für das vorherige MLM, mit dem Unterschied, daß diese Prüfung nach Dokumentation einer Histologie ausgelöst wird. Die gegebenenfalls generierten Meldungen sehen folgendermaßen aus :

- *Maligne Tumoren mit der/n Histologie/n ("||HIST_LISTE_D_TEXT||") sind bei Patienten unter 26 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich.*
- *Maligne Tumoren mit der/n Histologie/n ("||HIST_LISTE_D_TEXT||") sind bei Patienten über 14 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich.*

Die Variable HIST_LISTE_D_TEXT enthält eine Beschreibung der dokumentierten Histologie/n.

5.3.3.12. Verträglichkeit von Patientengeschlecht und Lokalisation eines Tumors

Diese Integritätsprüfung überprüft bei geschlechtsspezifischen Lokalisationen die Plausibilität zum Patientengeschlecht. Als typisch männliche Lokalisationen gelten hierbei die männlichen (Kategorien C60 bis C63 aus [WHO 1990]), als typisch weibliche Lokalisationen die weiblichen Genitalorgane (Kategorien C51 bis C58 aus [WHO 1990]). Wird bei einem weiblichen Patienten eine typisch männliche Lokalisationen, bzw. bei einem männlichen Patienten eine typisch weibliche Lokalisation ins GTDS eingegeben, so werden folgende Meldungen generiert :

- *Sie haben bei Herrn "||pat_name||" eine typisch weibliche Lokalisation ("||lok_code||") dokumentiert. Bitte kontrollieren ! (IARC-Integritätsprüfung)*
- *Sie haben bei Frau "||pat_name||" eine typisch männliche Lokalisation ("||lok_code||") dokumentiert. Bitte kontrollieren ! (IARC-Integritätsprüfung)*

Die Variable lok_code enthält eine Liste der zum Patienten dokumentierten Lokalisationen. Ausgelöst wird die Integritätsprüfung unmittelbar nach Dokumentation (Neueingabe, Modifikation) einer Lokalisation. Es wird als Annahme vorausgesetzt, daß zu diesem Zeitpunkt die Patientenstammdaten (einschließlich Patientengeschlecht) bereits dokumentiert wurden. Da eine Integritätsverletzung mit hoher Sicherheit auf einen Fehler hinweist, wird eine entsprechende Meldung als Fehler eingestuft. Das realisierte MLM enthält im prozeduralen Code der Logic-Slots das Fachwissen darüber, welche Lokalisationen typisch weiblich bzw. männlich ist. Da der Umfang des Fachwissens sich in Grenzen hält, wurde auf eine zentrale Ablage außerhalb eines MLMs in deklarativer Form verzichtet.

5.3.3.13. Verträglichkeit von Patientengeschlecht und Histologie eines Tumors

Diese Integritätsprüfung überprüft bei geschlechtsspezifischen Histologien die Plausibilität zum Patientengeschlecht. Die als typisch männlich bzw. weiblich geltenden Histologien sind aufgrund des Umfangs nicht an dieser Stelle angegeben. Sie können entweder in [Parkin 1994] oder in MLM iarc6 im Anhang nachgelesen werden. Wird bei einem weiblichen Patienten eine typisch männliche Histologie, bzw. bei einem männlichen Patienten eine typisch weibliche Histologie ins GTDS eingegeben, so werden folgende Meldungen generiert:

- *Sie haben bei Herrn "||pat_name||" eine typisch weibliche Histologie ("||hist_liste||") dokumentiert. Bitte kontrollieren ! (IARC-Integritätsprüfung)*
- *Sie haben bei Frau "||pat_name||" eine typisch männliche Histologie ("||hist_liste||") dokumentiert. Bitte kontrollieren ! (IARC-Integritätsprüfung)*

Die Variable `hist_liste` enthält eine Liste der zum Patienten dokumentierten Histologien. Ausgelöst wird die Integritätsprüfung unmittelbar nach Dokumentation (Neueingabe, Modifikation) einer Histologie. Es wird als Annahme vorausgesetzt, daß zu diesem Zeitpunkt die Patientenstammdaten (einschließlich Patientengeschlecht) bereits dokumentiert wurden. Da derartige Integritätsverletzung von der IARC lediglich als falsch oder unwahrscheinlich eingestuft sind, wird eine entsprechende Meldung als Warnung ausgegeben.

5.3.3.14. Plausibilität von Histologie- und Lokalisationsangaben

5.3.3.14.1. Strukturierung des verwendeten Fachwissens gemäß [Parkin 1994]

Das für die Plausibilität von Histologie und Lokalisationsangaben verwendete Fachwissen entspricht dem des in [Parkin 1994] publizierten Fachwissens und wurde uns auf Anfrage von der IARC in einer EDV-verwertbaren Form zur Verfügung gestellt.

Im einzelnen sieht die Strukturierung folgendermaßen aus

- Auflistung von Histologien, die mit allen Lokalisationen pauschal verträglich sind. Es handelt sich hierbei um 153 Histologien.
- Gruppierung der Histologien zu 62 histologischen Familien, die in Abbildung 24 dargestellt sind. Im Mittel sind jeder histologischen Familie ca. 8 Histologien zugeordnet.
- Auflistung von verträglichen Lokalisationen zu den ersten 59 histologischen Familien. Im Schnitt sind dies 25 Lokalisationen pro histologischer Familie.
- Auflistung von unverträglichen Lokalisationen zu den letzten drei histologischen Familien. Im Schnitt sind dies 30 Lokalisationen pro Familie. Implizit sind alle Lokalisationen zu einer

Histologie dieser histologischen Familien verträglich, wenn sie nicht in der entsprechenden Liste erscheinen.

1	Skin tumours	32	Breast tumours
2	Melanomas	33	Thyroid gland tumours
3	P.T.C.C.	34	Salivary gland tumours
4	Ductal neoplasms	35	Thymus tumours
5	Paraganglia (I)	36	Germ cell tumours (I)
6	Miscellaneous soft tissue tumours	37	Paraganglia (II)
7	Synovial sarcomas	38	Paraganglia (III)
8	Mesothelioma	39	Eye melanomas
9	Kaposi's sarcoma	40	Periosteal fibrosarcoma
10	Nerve tumours	41	Endometrial stromal sarcoma
11	Gastrinoma	42	Liver tumours (II)
12	Germinoma	43	Bone tumours
13	Papillary cystadenocarcinoma	44	Ewing's sarcoma
14	Choriocarcinoma and teratoma	45	Pineal gland tumours
15	Lobular carcinoma	46	Brain tumours
16	Embryonal sarcoma	47	Retinoblastoma
17	Choriocarcinoma	48	Nose tumours
18	Transitional cell carcinoma	49	Meningeal tumours
19	Neuroblastoma	50	Plasma cell tumours
20	Cloacogenic carcinoma	51	Leukaemia
21	Pancreas tumours	52	Ovarian tumours
22	Liver tumours (I)	53	Endolymphatic stromal myosis
23	Colon/rectum tumours	54	Long bone tumours
24	Lung tumours	55	Hydatidiform mole
25	Pituitary tumours	56	Myelofibrosis
26	Kidney tumours (I)	57	I.P.S.I.D.
27	Parathyroid gland tumours	58	Paraganglia (IV)
28	Follicular carcinoma	59	Intraepithelial neoplasia
29	Kidney tumours (II)	X1	Epithelial (and other) tumours
30	Suprarenal gland tumours	X2	Germ cell tumours (II)
31	Endometrioid carcinoma	X3	Giant cell sarcoma

Abb. 24 : Histologische Familien nach [IARC 1994]

5.3.3.14.2. Verträglichkeitskontrolle einer gerade dokumentierten Histologie mit bereits dokumentierten Lokalisationen unter Verwendung von *möglichen* Kombinationen

Diese Integritätsprüfung kontrolliert die Konsistenz einer dokumentierten Histologie mit den vorhandenen Lokalisationen zum Tumorfall. Hierbei wird auf zentral in relationalen Tabellen deklarativ abgelegten Fachwissen über die Zugehörigkeit histologischer Codes zu Familien und die akzeptierten Lokalisationsschlüssel zu den einzelnen histologischen Familien zurückgegriffen. Gründe für den Rückgriff auf zentral abgelegtes Fachwissen können in Kapitel 4.5.4. nachgelesen werden.

Die Prüfung soll folgenden Anforderungen gerecht werden :

- keine einschränkenden Annahmen über Anzahl von Histologie und Lokalisation

- Meldung soll zur Vermeidung einer „Meldungsinflation“ nur erfolgen, wenn die aktuell dokumentierte Histologie aufgrund des IARC-Wissens problematisch ist
- Meldungstext soll inhaltlich auf die aktuell dokumentierte Histologie bezogen sein.

Der konkrete Ablauf der Integritätsprüfung sieht folgendermaßen aus. Nach unmittelbarer Auslösung durch Dokumentation (Neueingabe, Modifikation) eines Histologieschlüsselwertes (Ereignisspezifität von Typ 3, 4) werden vereinfachend dargestellt zwei Lokalisationslisten aufgebaut. Die erste enthält alle dokumentierten Lokalisationen zum Tumorfall, die zweite alle zur dokumentierten Histologie möglichen Lokalisationen. Bei der Generierung der zweiten Liste wird auf das zentral abgelegte Faktenwissen über Familienzugehörigkeit und verträglichen Lokalisation zugegriffen. Nun wird überprüft, ob mindestens eine der dokumentierten Lokalisationen mit einer der möglichen Lokalisationen übereinstimmt. Konkret wird dies durch eine Schnittmengenbestimmung der beiden Listen festgestellt. Ist dies nicht der Fall, so wird die Meldung *„Nach den Empfehlungen der IARC ist bei `||anrede||` `||pat_name||` die Histologie (Schlüssel: `||hist_code||` aus Familie: `||HIST_FAMILIE_NAME||`) mit keiner der dok. Lokalisationen (Schlüssel: `||lok_liste_d||`) verträglich ! (Prüfung IARC10)“* generiert. Die Variable `hist_code` enthält den gerade dokumentierten Histologiewerte, die Variable `HIST_FAMILIE_NAME` den Namen der zugehörigen histologischen Familie und `lok_liste_d` alle dokumentierten Lokalisationen.

Eine bezüglich der Meldung von Fehlern empfindlichere Realisierungsalternative dieser Prüfung könnte so aussehen, daß schon dann eine Meldung generiert wird, wenn eine der dokumentierten Lokalisationen nicht einer möglichen entspricht. Das realisierte MLM zeigt die Bedeutung des Listen-Datentypes innerhalb der Arden-Syntax, ohne den eine solche Prüfung bedeutend aufwendiger zu realisieren wäre.

5.3.3.14.3. Verträglichkeitskontrolle einer gerade dokumentierten Histologie mit bereits dokumentierten Lokalisationen unter Verwendung von *sehr unwahrscheinlichen* Kombinationen

Diese Integritätsprüfung kontrolliert wie auch schon die vorherige Integritätsprüfung die Konsistenz einer dokumentierten Histologie mit den vorhandenen Lokalisationen zum Tumorfall. Sie unterscheidet sich von der vorherigen Prüfung dadurch, daß sie auf das IARC-Fachwissen über *sehr unwahrscheinliche* Lokalisationen zu histologischen Familien zurückgreift. Der Grund für diese auf den ersten Blick überraschende Strukturierung des Wissens zur Konsistenz von Lokalisation und Histologie ist rein pragmatischer Natur : für

einige histologische Familien ist es leichter, verträgliche Lokalisationen aufzulisten, während es für andere Familien leichter ist, unverträgliche Lokalisationen zu sammeln. Während der Ableitung werden zwei Lokalisationslisten aufgebaut, wobei die eine die dokumentierten Lokalisationen, die andere basierend auf dem IARC-Fachwissen alle zur auslösenden Histologie sehr unwahrscheinlichen Lokalisationen enthält. Die Meldung *"Nach den Empfehlungen der IARC ist bei `||anrede||` `||pat_name||` die Histologie (Schlüssel: `||hist_code||` aus Familie: `||HIST_FAMILIE_NAME||`) mit mindestens einer der dok. Lokalisationen (Schlüssel: `||lok_liste_d||`) UNverträglich ! (Prüfung IARC11)"* wird generiert, wenn eine der dokumentierten Lokalisationen einer der sehr unwahrscheinlichen Lokalisationen entspricht. Die Variable `hist_code` enthält den gerade dokumentierten Histologiewert, die Variable `HIST_FAMILIE_NAME` den Namen der zugehörigen histologischen Familie und `lok_liste_d` alle dokumentierten Lokalisationen.

5.3.4. Umgebung zum nachträglichen Überarbeiten von Dokumenten

Die Arden-Shell ergänzt um die besprochenen MLMs weist den Benutzer bei der Erfassung auf die Verletzung von Integritätsbedingungen hin. Durch die erfassungsnahe Meldung wird es dem Benutzer erleichtert, geeignete Maßnahmen zur Korrektur zu ergreifen, da er sich in den entsprechenden GTDS-Eingabemasken befindet und das als Datenquelle dienende Originaldokument (Bsp. Krankenakte) vorliegen hat. Für den Fall, daß der Benutzer ein Eingehen auf die Meldung auf einen späteren Zeitpunkt verschieben möchte, wurde eine eigenständige Umgebung geschaffen, die in Abbildung 25 abgebildet ist.

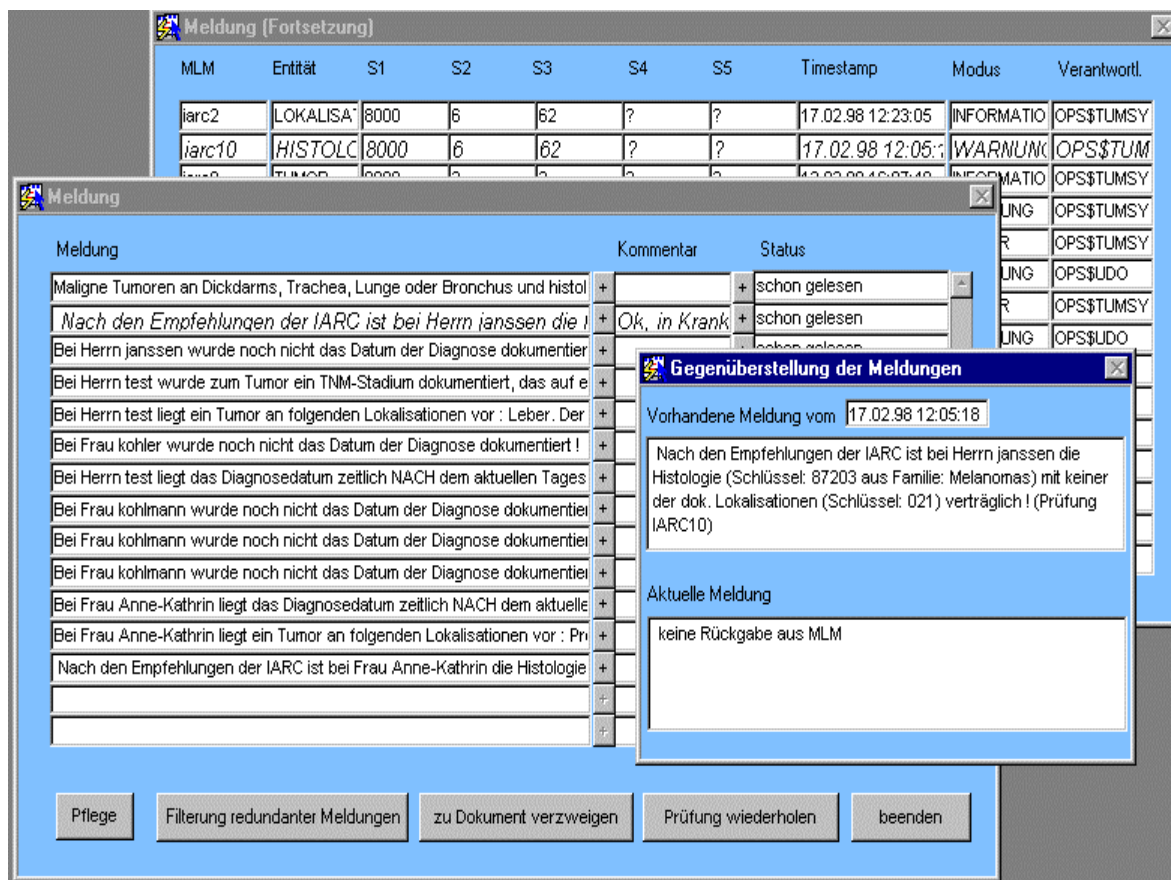


Abb. 25: Arbeitsumgebung zur nachträglichen Überarbeitung von Dokumenten

Neben einer übersichtlichen Darstellung aller Meldungen hat der Benutzer die Möglichkeit, die einer Meldung zugrundeliegende Integritätsprüfung zu wiederholen. Dies wurde in Abbildung 25 für die kursiv geschriebene Meldung durchgeführt. Aus dem Fenster „Gegenüberstellung der Meldungen“ kann man entnehmen, daß der aktuelle Aufruf zu keinem Meldungshinweis geführt hat. Dies weist darauf hin, daß der Verstoß gegen die entsprechende Integritätsbedingung bereinigt wurde. Eine andere Möglichkeit der Umgebung ist das gezielte Verzweigen zum Dokument, auf das sich eine Meldung bezieht, um ggf. einen Fehler zu bereinigen.

5.4. Anwendungsfeld: Studienzuteilung von Patienten

Da im Zusammenhang mit onkologischen Informationssystemen ein ständiger Wunsch nach Unterstützung bei der Auswahl geeigneter Patienten für klinische Studien besteht, soll in diesem Kapitel eine diesbezügliche Verwendung der Shell für wissensverarbeitende Funktionen anhand eines Beispiels vorgestellt werden.

5.4.1. Konzeption der Unterstützung

Kernstück der Konzeption ist ein allgemeiner Scoring-Algorithmus, der zu einem Tumorfall eine Einschätzung über die Eignung zu einer Studien macht. Der Algorithmus vergibt für jedes berücksichtigte Ein- bzw. Ausschlußkriterium Punkte. Hierauf basierend wird letztendlich unterschieden, ob ein Patient für eine konkrete Studie

- geeignet,
- definitiv nicht geeignet oder
- eventuell bzw. noch nicht geeignet ist.

Der letzte Fall tritt dann ein, wenn Kriterien mangels relevanter Daten nicht bewertet werden können oder aber erfüllte Ausschlußkriterien nicht endgültiger Natur sind, wie dies beispielsweise bei einem geforderten bestimmten Leistungszustand oder der Nichtdurchführung einer Chemotherapie in den letzten zwei Monaten der Fall sein könnte. Jede Einschätzung endet mit einer Auflistung der berücksichtigten Kriterien und deren Bewertung hinsichtlich der getroffenen Entscheidung. Beispiele für solche Einschätzungen können aus den Abbildungen 26 und 27 entnommen werden. Dieser Ansatz führt zu einem MLM für jede abzuprüfende Studie. Im nächsten Kapitel wird dieser Algorithmus anhand eines realisierten MLMs erläutert. Bezüglich der Anwendung der resultierenden MLMs gibt es zwei Strategien, die sich dadurch voneinander unterscheiden, ob sie während der Dokumentation patientenorientiert („Online-Strategie“) oder aber als unabhängige Tätigkeit für ein Patientenkollektiv („Offline-Strategie“) ablaufen.

1. „Online-Strategie“: Hier erfolgt die Auslösung der Studien-MLMs zu einem Patienten über einen eigens hierfür geschaffenen Knopf „Studienzuweisung“ innerhalb der GTDS-Diagnosemaske mit Ausgabe der generierten Meldungen über die Meldungskomponente der Shell für wissensverarbeitende Funktionen. Diese Strategie erfordert im Gegensatz zur zweiten eine Modifikation am GTDS. Abbildung 26 zeigt eine Situation, wie sie sich bei einer Sitzung am GTDS ergeben kann.

The screenshot shows the 'DIAGNOSEDATEN' window with the following content:

- Patient Data:**
 - Patient: Janssen, Johann, 10.10.90
 - Abt.: INN, INNERE MEDIZIN, KRANKENHAUS NR.1, Gießen
 - Tumor: wider testdeiangosne
 - ID.: 8000
 - ID.: 1
 - ID.: 3
- Message Box: ***INFORMATION aus dem ARDEN-Modul*****
 - Studie03:Herr Janssen erfüllt NICHT die Kriterien (jünger als 18(Ausschluss), keine Chemotherapie(OK), keine Hirnmetastasen (OK), Pankreaskarzinom(OK))
 - Buttons: OK, Abbrechen
- HISTOLOGIE Table:**

HN	Code	Beschreibung	Grading	Datum	Auf.
Neben	81513	häufigste INSULINOM, BÖSARTIGES	1		2
- ICD-9 and ICD-10 Fields:**
 - ICD-9: Beschreibung [Empty]
 - ICD-10: Beschreibung [Empty]
- Buttons:** Studienzuweisung, zurück zur Hauptseite
- Status Bar:** Kontrolle auf Eignung zu vorhandenen Studien ... Count: *1

Abb.26 : „Online“-Anwendungsstrategie am graphischen GTDS

2. „Offline-Strategie“: Hier erfolgt die Auslösung der Studien-MLMs zu allen Patienten und Präsentation der Meldungen aus einer eigenen Benutzeroberfläche heraus. Um diese Strategie zu ermöglichen, wurde bei der Namensgebung der MLMs, sowie bei den generierten Meldungen auf ein festes Format Wert gelegt. So beginnen die Namen aller Studien-MLMs mit „ST“ (Beispiel „ST_SMSGEM1“) und alle generierten Meldung mit dem Namen der Studie (Beispiel „STUDIE03:Frau Meyer noch unvollständig ...“). Diese „Offline“-Anwendungsstrategie dürfte bei diesem Anwendungsfeld wissensverarbeitender Funktionen die aus Anwendersicht die bessere sein, da so die Studienzuteilungstätigkeit separat von der Dokumentation durchgeführt werden kann. In Abbildung 27 wird eine entsprechende Oberfläche dargestellt, die sämtliche Meldungen zu den ausgewählten Tumorfällen enthält.

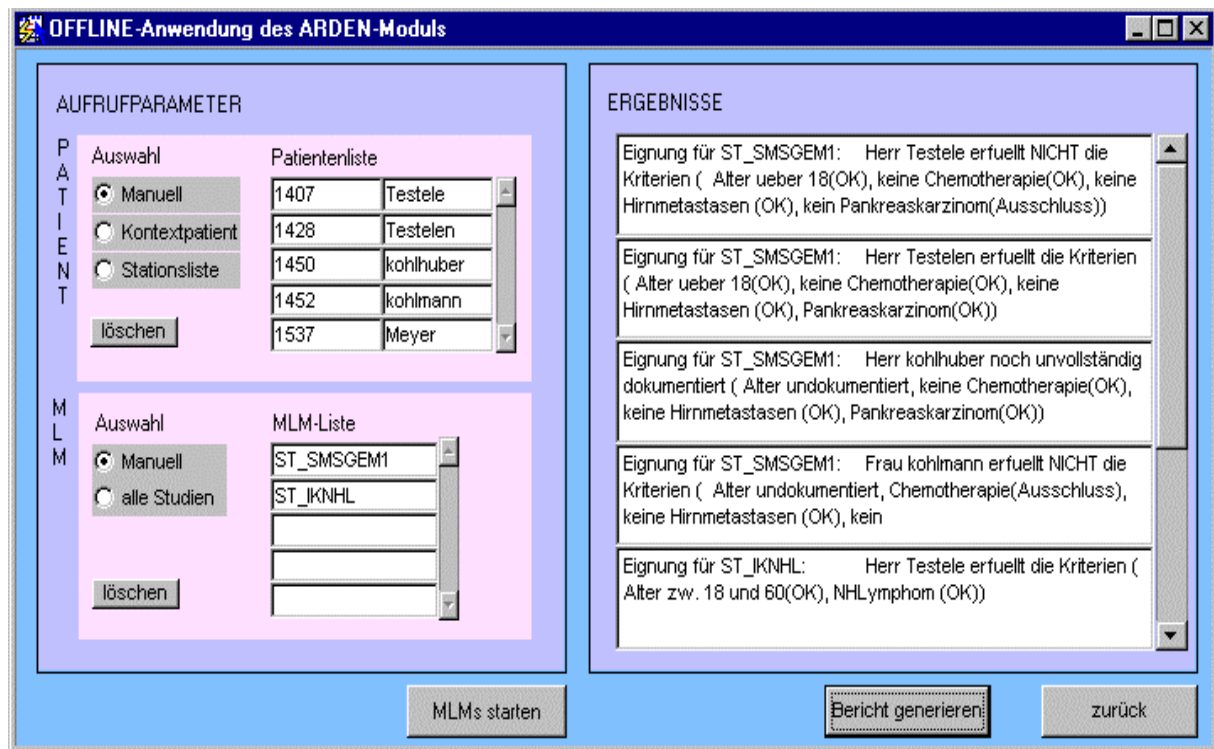


Abb.27 :Benutzeroberfläche für die „Offline“-Anwendungsstrategie

In der linken Hälfte der Oberfläche kann man die zu berücksichtigenden Patienten und Studien spezifizieren. Basierend hierauf wird nach Aktivieren des Knopfes „MLMs starten“ die rechts angezeigte Liste generiert. Hierbei handelt es sich bei jedem Eintrag um die Meldung eines MLMs zur Eignung eines Patienten zu einer Studie. Das festgelegte Format der Meldungen erlaubt es, diese als Ausgangsbasis für weitere Verarbeitung, wie z.B. einer Berichtserstellung mit detaillierten Angaben zu Patient und in Frage kommenden Studien, zu verwenden.

5.4.2. Ein MLM zur Unterstützung einer Studienzuteilung

Das realisierte MLM führt eine Zuteilungsunterstützung zu einer Phase II Studie zur Untersuchung der Verträglichkeit und Wirksamkeit von hochdosiertem Octreotid in Kombination mit Gemcitabin zur Therapie des lokal fortgeschrittenen oder metastasierten Pankreaskarzinoms durch. Es wurden exemplarisch je zwei Ein- und Ausschlusskriterien berücksichtigt :

- Einschlusskriterium Alter mindestens 18 Jahre
- Einschlusskriterium histologisch nachgewiesenes oder klinisch eindeutig belegtes Pankreaskarzinom mit Fernmetastasen oder lokal fortgeschrittener

Erkrankung

- Ausschlusskriterium durchgeführte Chemotherapie
- Ausschlusskriterium bekannte ZNS-Metastasen

Die folgende Abbildung 28 zeigt die Knowledge-Kategorie des realisierten MLMs.

```
DATA:
UPDATE_TUMOR_TRICK:= EVENT {update,tumor,aenderungsdatum};
HISTOLOGIE_LISTE := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient =
st_smsgem1.fk_tumorfk_patient and fk_tumortumor_id = st_smsgem1.fk_tumortumor_id};
geburtsdatum := READ last{SQL: select geburtsdatum from patient where pat_id = st_smsgem1.fk_tumorfk_patient};
anz_chemotherapie := READ count {SQL: select fk_patientpat_id from internistische_therapie where fk_patientpat_id =
st_smsgem1.fk_tumorfk_patient and fk_tumortumor_id = st_smsgem1.fk_tumortumor_id};
anz_metastase_bra := READ count {SQL: select fk_tumorfk_patient from metastase where fk_tumorfk_patient =
st_smsgem1.fk_tumorfk_patient and fk_tumortumor_id = st_smsgem1.fk_tumortumor_id and fk_lokalisierung = 'BRA'};
pat_name := READ last{SQL: select name from patient where pat_id = st_smsgem1.fk_tumorfk_patient};
pat_geschl:= READ last{SQL: select geschlecht from patient where pat_id = st_smsgem1.fk_tumorfk_patient};
;;
EVOKE:
UPDATE_TUMOR_TRICK;
;;
LOGIC:
if (pat_geschl = "M") then
    anrede := "Herr";
else
    anrede := "Frau";
endif;

/*Variablen und Zaehler*/
flag_unerfuellt := false;
flag_erfuellt := false;
flag_womoeglich := false;
text2 := "Beschreibung der Bedingungen";
zaehler_unerfuellt := 0; /*Zaehler fuer unerfuellte Bedingungen*/
zaehler_erfuellt := 0; /*Zaehler fuer erfuellte Bedingungen*/
zaehler_womoeglich := 0; /*Zaehler fuer unentschiedene Bedingungen*/

/*1. Bedingung : Alter mindestens 18*/
age:=alter(geburtsdatum);
if ((age < 18) and (age > 0)) then
    text2 := "jünger als 18(Ausschluss)";
    zaehler_unerfuellt := zaehler_unerfuellt + 1;
else
    if (age > 17) then
        text2 := "Alter ueber 18(OK)";
        zaehler_erfuellt := zaehler_erfuellt + 1;
    else
        text2 := "Alter undokumentiert";
        zaehler_womoeglich := zaehler_womoeglich + 1;
    endif;
endif;

/*2. Bedingung : keine Chemotherapie*/
if (anz_chemotherapie >0)
then
    text2:= text2||", Chemotherapie(Ausschluss)";
    zaehler_unerfuellt := zaehler_unerfuellt + 1;
else text2:= text2||", keine Chemotherapie(OK)";
    zaehler_erfuellt := zaehler_erfuellt + 1;
endif;

/*3. Bedingung : keine Hirnmetasen*/
if (anz_metastase_bra >0)
then
    text2:=text2||", Hirnmetastase(Ausschluss)";
    zaehler_unerfuellt := zaehler_unerfuellt + 1;
else text2:=text2||", keine Hirnmetastasen (OK)";
    zaehler_erfuellt := zaehler_erfuellt + 1;
endif;

/*4. Bedingung : eine der Histo. 8152,8151,8153 liegt vor*/
histologie_liste:= substr(histologie_liste,1,4); /*nun Dignitaet weg*/
if ((histologie_liste = "8152") or
```

```

(histologie_liste = "8151") or
(histologie_liste = "8153"))
then text2:=text2||", Pankreaskarzinom(OK)";
  zaehler_erfuellt := zaehler_erfuellt + 1;
else
  text2:=text2||", kein Pankreaskarzinom(Ausschluss)";
  zaehler_unerfuellt := zaehler_unerfuellt + 1;
endif;

/*Endeverarbeitung*/
if (zaehler_unerfuellt > 0) then
  flag_unerfuellt := true;
  meldungs_modus := "WARNUNG";
  conclude true;
endif;
if (zaehler_erfuellt = 4) then
  flag_erfuellt := true;
  meldungs_modus := "WARNUNG";
  conclude true;
endif;
if (zaehler_womoeglich + zaehler_erfuellt = 4) then
  flag_womoeglich := true;
  meldungs_modus := "WARNUNG";
  conclude true;
endif;
::
ACTION:
if flag_unerfuellt then
WRITE "Studie03:"||anrede||" "||pat_name||" erfuehlt NICHT die Kriterien ( "||text2||")";
endif;

if flag_erfuellt then
WRITE "Studie03:"||anrede||" "||pat_name||" erfuehlt die Kriterien ( "||text2||")";
endif;

if flag_womoeglich then
write "Studie03:"||anrede||" "||pat_name||" noch unvollständig dokumentiert ( "||text2||")";
endif;
::

```

Abb. 28 : Knowledge Kategorie des Beispiel-MLMs zur Unterstützung des Studienmanagements

Zentrale Bedeutung haben die drei Variablen `zaehler_unerfuellt`, `zaehler_erfuellt` und `zaehler_womöglich`. Je nachdem, ob aus den aus dem GTDS gelesenen Daten auf negative oder positive Eignung des Patienten für die Studie geschlossen werden kann, werden die Variablen `zaehler_unerfuellt`, `zaehler_erfuellt` um eins erhöht. Kann aus den gelesenen Daten kein positiver oder negativer Schluß gezogen werden, wie zum Beispiel bei Nichtvorhandensein eines Geburtstages, so wird die Variable `zaehler_womöglich` hochgezählt. Der Scoring-Algorithmus wird hierdurch in der Lage versetzt, mögliche Studienkandidaten - trotz fehlender Daten oder Erfüllung von Ausschlußkriterien, die jedoch nicht zwangsläufig zu einem endgültigen Ausschluß führen - als solche zu erkennen. Parallel zu den Kriterienabprüfungen wird in der Variable `text2` eine Beschreibung der Kriterienbewertungen abgelegt.

Eine mögliche Meldung sieht folgendermaßen aus :

```
Studie03:Herr Janssen erfuehlt NICHT die
Kriterien
(jünger als 18 (Ausschluss),
Chemotherapie (Ausschluss),
keine Hirnmetastasen (OK),
kein Pankreaskarzinom (Ausschluss))
```

Die letzten vier Zeilen enthalten hierbei den Wert der Variable `text2`. Zu jedem dokumentierten Wert ist angegeben, welche Rolle er in der Bewertung der Eignung des Patienten zur Studie spielt. Aus der Meldung des Scoring-Algorithmuses kann also gegebenenfalls entnommen werden, wie „weit“ ein Patient von der Eignung für eine Studie entfernt ist. Dies ist insbesondere bei Kriterien, deren Bewertung sich verändern kann, von Bedeutung.

5.4.3. Zusammenfassung

Der realisierte Scoring-Algorithmus erlaubt durch die Art, wie er mit fehlenden Daten oder Ausschlusskriterien umgeht, deren Ausprägung für die Entscheidung von nicht endgültiger Natur sind, eine relativ differenzierte Einschätzung über die Eignung eines Patienten zur Teilnahme an einer Studie. Die Entscheidung genau eine Studienentscheidung in einem MLM abzulegen, ermöglicht eine flexible Anpassung an sich ändernde Studienbedingungen: abgelaufene Studien können einfach durch Löschen des entsprechenden MLMs aus dem System entfernt werden. Bei der Berücksichtigung neuer Studien kann die Struktur des Scoring-Algorithmus wiederverwendet werden und muß lediglich an die neuen Kriterien angepaßt werden.

Die optimale Anwendungsstrategie scheint im Gegensatz zum Anwendungsgebiet „Sicherung der Datenqualität“ eine „Offline“-Strategie zu sein.

6. Klinischer Einsatz im Tumorzentrum Cottbus

Seit Anfang Mai 1998 wird die Shell für wissensverarbeitende Funktionen - vervollständigt um die vorgestellten MLMs zum Anwendungsfeld „Sicherung von Datenqualität“ aus Abbildung 21 - im Tumorzentrum Cottbus zur *erfassungsnahen* Unterstützung des dokumentierenden Personals bei der Erkennung von Unplausibilitäten und Unvollständigkeiten eingesetzt. Die Unterstützung besteht im unmittelbaren Aufblenden eines Hinweises durch den Arden-Melder analog zu Abbildung 14 im Falle der Dokumentation eines Sachverhaltes, der gegen eine der in den MLMs zugrundeliegenden Integritätsbedingungen verstößt. Durch die Shell selber werden keine Benutzereingaben am GTDS blockiert oder Daten verändert; es bleibt dem Benutzer überlassen, adäquat auf die Hinweise zu reagieren.

6.1. Möglichkeit zum Bewerten und Nachvollziehen der Systemaktivität

Um die Aktivität des Systems sowie die Benutzerreaktionen nachvollziehen zu können, wurde die Shell um folgende Funktionen ergänzt :

- um eine Aufforderung an Benutzer, eine gerade gelesene Meldung mittels einer vorgegebenen Werteliste zu beurteilen. Abbildung 29 zeigt die aktuelle vor Ort verwendete Oberfläche des Arden-Melders mit der integrierten Bewertungsfunktionalität, wie sie sich einem Benutzer unmittelbar nach der Meldungspräsentation zeigt.



Abb. 29: Arden-Melder mit aufgeblendeter Bewertungsliste

- um die Möglichkeit, Informationen über die Aktivität der Shell zu Auswertungszwecken zu exportieren. Diese Informationen umfassen Angaben über die Shellaktivität (entdeckte Ereignisse, generierte Meldungen) sowie über die Beurteilung durch den Benutzer. Abbildung 30 und 31 zeigen Ausschnitte aus konkreten Exportdateien.

EREIGNIS:	INSERT_HISTOLOGIEIARC6	DATUM:	24.06.1998	ANZAHL:	23
EREIGNIS:	INSERT_LOKALISATIONIARC13	DATUM:	24.06.1998	ANZAHL:	13
EREIGNIS:	INSERT_LOKALISATIONIARC2	DATUM:	24.06.1998	ANZAHL:	13
EREIGNIS:	INSERT_LOKALISATIONIARC3	DATUM:	24.06.1998	ANZAHL:	13
EREIGNIS:	INSERT_OP_DATUM	DATUM:	24.06.1998	ANZAHL:	13
EREIGNIS:	INSERT_TNM	DATUM:	24.06.1998	ANZAHL:	23
EREIGNIS:	INSERT_TUMOR	DATUM:	24.06.1998	ANZAHL:	12
EREIGNIS:	UPDATE_DIA_DATUM	DATUM:	24.06.1998	ANZAHL:	13
EREIGNIS:	UPDATE_HISTOLOGIEIARC10	DATUM:	24.06.1998	ANZAHL:	3
EREIGNIS:	UPDATE_HISTOLOGIEIARC11	DATUM:	24.06.1998	ANZAHL:	3
EREIGNIS:	UPDATE_HISTOLOGIEIARC2	DATUM:	24.06.1998	ANZAHL:	3
EREIGNIS:	UPDATE_HISTOLOGIEIARC4	DATUM:	24.06.1998	ANZAHL:	3
EREIGNIS:	UPDATE_HISTOLOGIEIARC6	DATUM:	24.06.1998	ANZAHL:	3
EREIGNIS:	UPDATE_LOKALISATIONIARC13	DATUM:	24.06.1998	ANZAHL:	1

Abb. 30 : Ausschnitt aus der Exportdatei mit Informationen über detektierte Ereignisse

Jede Zeile der Datei beschreibt, wie oft ein bestimmtes Ereignis an einem bestimmten Tag vorlag.

VERANT.: OPS\$WITTE MLM: iarc10	TEXT.: Nach den Empfehlungen der IARC ist bei Frau XXXXXX die Histologie (Schlüssel: 83303 aus Familie: Follicular carcinoma) mit keiner der dok. Lokalisationen (Schlüssel: 549) verträglich ! (Prüfung IARC10)	KONTEXT:	HISTOLOGIE 28124 1 1 ? ?	EVENT:	Event:INSERT_HISTOLOGIEIARC10	Aufruf:9.06.98 12:18:30	Datensatz:28124/1/1/?/?	BEW.: 1	STATUS: schon gelesen ID: 2953
VERANT.: OPS\$WITTE MLM: iarc13	TEXT.: Nach den Empfehlungen der IARC ist bei Frau XXXXXXXXXX die Lokalisation (Schlüssel: 548) mit KEINER der dok. Histologien (aus den Familien Ovarian tumours) verträglich ! (Prüfung IARC13 Fall 2)	KONTEXT:	LOKALISATION 27807 1 548 ? ?	EVENT:	Event:INSERT_LOKALISATIONIARC13	Aufruf:11.05.98 15:40:30	Datensatz:27807/1/548/?/?	BEW.:	STATUS: schon gelesen ID: 486
VERANT.: OPS\$WITTE MLM: iarc10	TEXT.: Nach den Empfehlungen der IARC ist bei Frau XXXXXXXXXX die Histologie (Schlüssel: 84413 aus Familie: Ovarian tumours) mit keiner der dok. Lokalisationen (Schlüssel: 548) verträglich ! (Prüfung IARC10)	KONTEXT:	HISTOLOGIE 27807 1 1 ? ?	EVENT:	Event:INSERT_HISTOLOGIEIARC10	Aufruf:11.05.98 15:41:12	Datensatz:27807/1/1/?/?	BEW.:	STATUS: schon gelesen ID: 487
VERANT.: OPS\$LEH MLM: iarc0	TEXT.: Bei Frau XXXXXXXXXX wurde noch nicht das Datum der Diagnose dokumentiert !	KONTEXT:	TUMOR 28286 1 ? ? ? ?	EVENT:	Event:INSERT_TUMOR	Aufruf:29.06.98 14:44:58	Datensatz:28286/1/?/?/?	BEW.:	STATUS: ID: 1475
VERANT.: OPS\$LEH MLM: iarc0	TEXT.: Bei Herrn XXXXXXXX wurde noch nicht das Datum der Diagnose dokumentiert !	KONTEXT:	TUMOR 28276 1 ? ? ? ?	EVENT:	Event:INSERT_TUMOR	Aufruf:29.06.98 11:42:3	Datensatz:28276/1/?/?/?	BEW.:	STATUS: rev.wg.MLM 1751 ID: 1398

Abb. 31 : Ausschnitt aus der Exportdatei mit Informationen über generierte Meldungen

Jeder der fünf Textblöcke in Abbildung 31 enthält Angaben zu einer generierten Meldung. Im wesentlichen sind dies :

- der verantwortliche GTDS-Benutzer,
- das meldungsgenerierende MLM,
- der komplette Meldungstext,
- das auslösende Ereignis,

- der Kontext der Meldung, der durch Angabe von Tabelle und Schlüsselwerten das auslösende Ereignis weiter beschreibt,
- der genaue Zeitpunkt des Aufrufes,
- die Bewertung durch den Benutzer und
- der Status, aus dem entnommen werden kann, ob das MLM schon gelesen oder durch den retract-Mechanismus der Arden-Engine als störend gekennzeichnet wurde.

Von den fünf Meldungen in Abbildung 31 wurden beispielsweise nur die ersten drei vom Benutzer gelesen. Die vierte Meldung wurde nicht gelesen, während die fünfte vom retract-Mechanismus als störend erkannt wurde. Weiterhin wurde die erste Meldung vom Benutzer als inhaltlich korrekt gekennzeichnet. Die anderen gelesenen Meldungen wurden nicht bewertet.

6.2. Erste Ergebnisse

Grundlage der ersten Ergebnisse ist das bis Ende Oktober durch die in Kapitel 6.1. beschriebene Exportfunktion gewonnene Datenmaterial.

6.2.1. Ergebnisse zum Meldungsaufkommen

Abbildung 32 zeigt die Anzahl aller generierten Meldungen differenziert nach den verantwortlichen MLMs. Es fällt das hohe Aufkommen von Meldungen zum MLM iarc0 und zum MLM ip3 auf.

Der überwältigende Anteil der Meldungen des MLMs iarc0 verweist auf eine Unvollständigkeit der Diagnosedokumentation („kein Diagnosedatum angegeben“). Hintergrund ist das konkrete Dokumentationsverhalten vor Ort : seit Mitte Juni legte eine Sekretärin bei jeder Neuaufnahme eines Patienten neben den Patientenstammdaten ein erstes Diagnosedokument an, um dort ausschließlich das Aufnahmedatum zu dokumentieren, was jeweils zum obengenannten Unvollständigkeitshinweis führte.

Die große Anzahl von Meldungen zum MLM ip3 ist auf eine irrtümlich eingespielte fehlerhafte Version des MLMs im Zeitraum vom 8. Juni bis zum 10. Juni zurückzuführen. Das damit korrelierte zeitlich verstärkte Meldungsaufkommen kann in Abbildung 34 nachvollzogen werden.

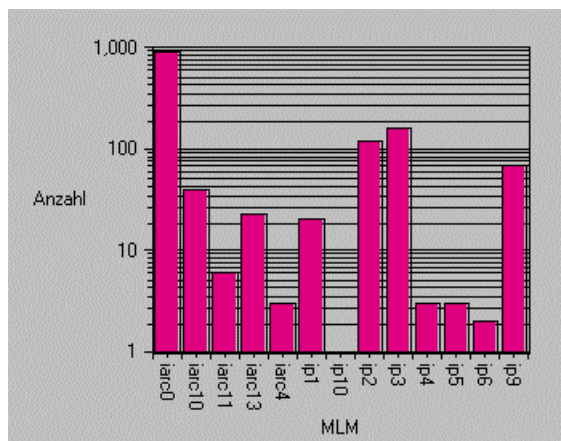


Abb. 32 : Anzahl aller generierten Meldungen

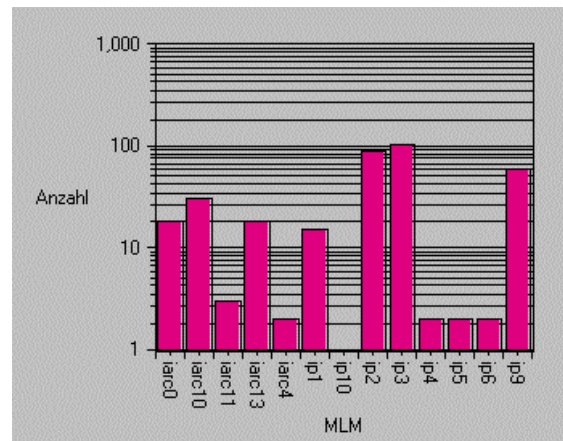


Abb. 33 : Anzahl der Meldungen ohne „retracted alerts“

Das Phänomen störender Meldungen bei automatisch ausgelösten wissensverarbeitenden Funktionen konnte auch bei diesem Einsatz beobachtet werden. Diese resultieren aus einer automatischen Auslösung der Funktionen verbunden mit einer nicht asynchronen Meldungspräsentation, wodurch beispielsweise eine Meldung zum Präsentationszeitpunkt schon veraltet sein kann. Diese Erfahrung deckt sich mit denen von [Kuperman 1997], der ein vergleichbares Phänomen innerhalb des Brigham Integrated Computing System (BICS) beschreibt.

durch Vergleich der Abbildungen 32 und 33 wird der Effekt des retract-Mechanismus deutlich, der der Ausblendung störender Meldungen für den Benutzer dient. Abbildung 33 unterscheidet sich von Abbildung 32 dadurch, daß die Meldungen, die durch den retract-Mechanismus der Shell als benutzerstörend gekennzeichnet wurden (und somit dem Benutzer nicht präsentiert werden), ausgeblendet sind. Die Meldungsanzahlen werden durch den retract-Mechanismus annähernd halbiert : allerdings variiert der Reduktionsgrad von Faktor 40 beim MLM iarc0 bis zu keiner Reduktion bei MLM ip6. Besonders auffällig ist die Abnahme der Meldungsanzahl zum MLM iarc0. Dies ist im konkreten Fall darauf zurückzuführen, daß der Hinweis auf Unvollständigkeit des Diagnosedokumentes („kein Diagnosedatum angegeben“) aufgrund des oft sehr schnellen Verlassens der Diagnosemaske nicht angezeigt wurde. Da beim nächsten Modifizieren des Dokumentes in den meisten Fällen ein gültiges Diagnosedatum dokumentiert wurde, wurde die ungelesene, nun inhaltliche veraltete Meldung vom retract-Mechanismus zurückgezogen.

Abbildung 34 zeigt die Verteilung der Meldungsanzahlen über den gesamten Zeitraum. Es kann eine Reduktion des Meldungsaufkommens einige Tage nach Systemeinführung erkannt werden. Der Hintergrund ist, daß

- einige MLMs aus dem System entfernt wurden - hierzu gehört das MLM ip1 („Kontrolle der Anzahl von Haupthistologien zu einem Tumorfall“) und das MLM ip4 („Kontrolle der Verlaufshistologien mit vorhandenen Histologien“) -
- ein Fehler im Ableitungsalgorithmus des MLM iarc13 (Lokalisation-Histologie-Verträglichkeit) aufgedeckt wurde. Der Fehler bestand darin, daß das indirekt im IARC-Fachwissen enthaltene Wissen, daß eine Lokalisation mit einer Histologie einer histologischen Familie verträglich ist, wenn diese Lokalisation nicht in der Liste der unverträglichen Lokalisationen enthalten ist, nicht berücksichtigt wurde.

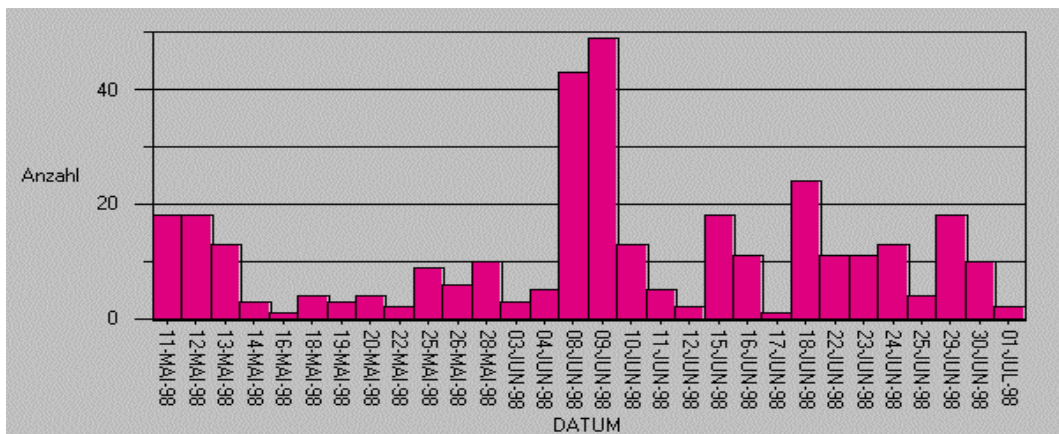


Abb. 34 : Verteilung aller Meldungen über den Zeitraum

Abbildung 35 blendet im Gegensatz zu Abbildung 34 die „retracted alerts“ aus. Sieht man vom Größenmaßstab ab, so ähneln sich die Verläufe der beiden Diagramme. Auffallend ist eine größere Differenz der Verläufe ab Mitte Juni. Dies ist, wie schon weiter oben angesprochen, auf das verstärkte unvollständige Anlegen von Diagnosedokumenten ab diesem Zeitpunkt zurückzuführen.

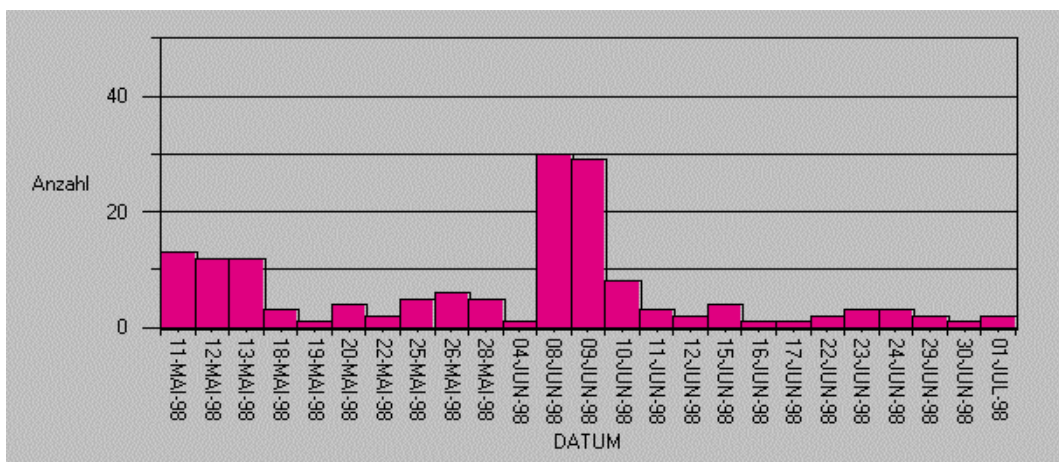


Abb. 35 : Verteilung aller Meldungen (ohne „retracted alerts“) über den Zeitraum

Um die vorhandenen Meldungen mit dem eigentlichen Dokumentationsaufkommen in Verbindung setzen zu können, wird seit dem 24. Juni ebenfalls die Anzahl der aufgetretenen für die Shell relevanten Ereignisse festgehalten. Abbildung 36 zeigt hierzu die ersten Zahlen. Etwa eins von hundert aufgerufenen MLMs generierte eine Meldung.

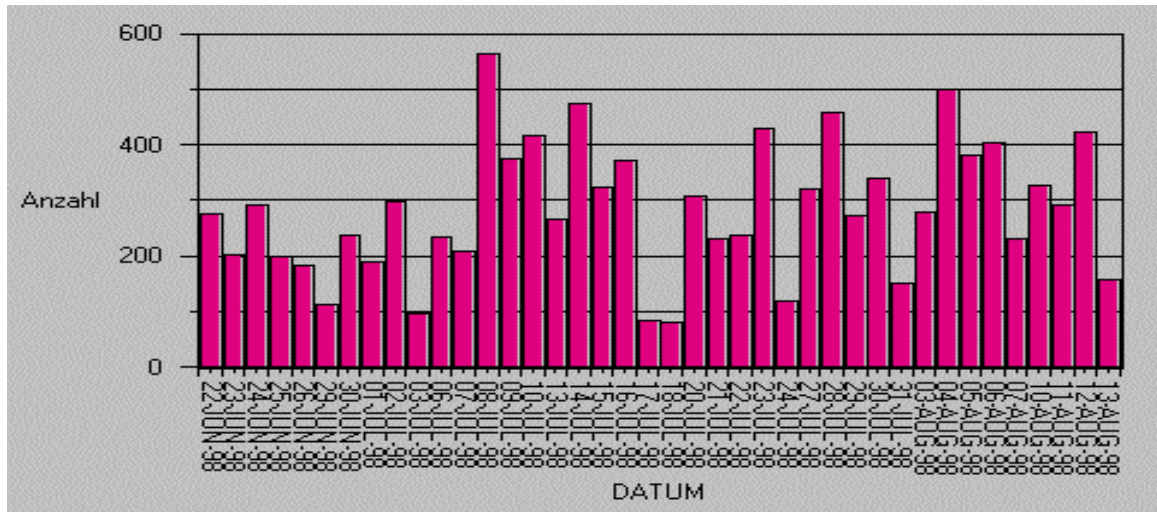


Abb. 36 : Verteilung registrierter Ereignisse über die Zeit (begonnen ab 24. Juni 1998)

6.2.2. Bewertung durch den Benutzer

Die folgenden Abbildungen spiegeln die Bewertung der Meldungen durch die Benutzer wieder. Nach jeder präsentierten Meldung wird der Benutzer zu einer standardisierten Beurteilung analog Abbildung 29 aufgefordert. Bis zum 21.5.1998 sah diese folgendermaßen aus :

1. inhaltlich OK, Anstoß für Fehlerbereinigung
2. wie oben, keine Fehlerbereinigung
3. kann nicht beurteilt werden
4. inhaltlich nicht OK
5. störende Meldung

Hintergrund dieser vorgegebenen Werte war es, einen Anhaltspunkt

- über die inhaltliche Einschätzung des Hinweises durch den Benutzer und
- über die Beeinflussung des Dokumentationsverhaltens

zu bekommen. Die Beurteilung wurde jedoch von einigen Dokumentationskräften als zu kompliziert eingestuft. Dies erklärt möglicherweise auch den hohen Anteil unbewerteter Meldungen und die Nichtverwendung der Bewertungsmöglichkeiten 2 und 3. Die Bewertung

nach der genannten Werteliste ist in Abbildung 37 dargestellt. Die letzte Säule stellt die Anzahl der unbewerteten Meldungen dar.

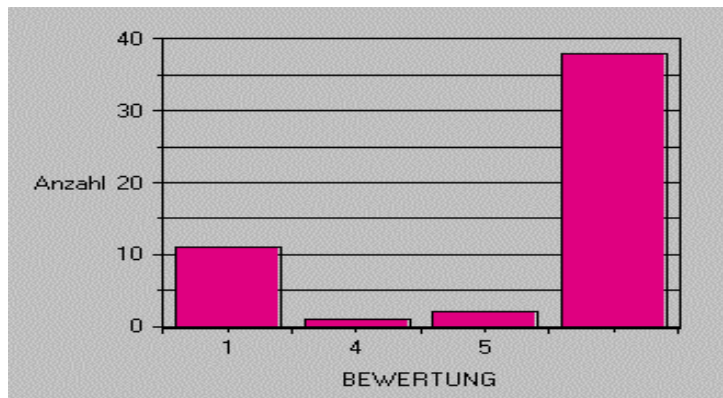


Abb. 37 : Meldungsbewertung durch den Anwender bis zum 21.5.1998

Ab dem 21.5. wurde eine vereinfachte Beurteilung mit folgenden Möglichkeiten durchgeführt:

1. inhaltlich OK
2. kann (noch) nicht beurteilt werden
3. inhaltlich falsch

Die Aussagekraft möglicher Beurteilung beschränkt sich auf die inhaltliche Einschätzung der entsprechenden Meldung durch den Benutzer.

Die Abbildung 38 zeigt die Bewertung der gelesenen Meldungen seit dem 22.5.1998. Die letzte Säule stellt -analog zu Abbildung 37- die unbewerteten Meldungen dar.

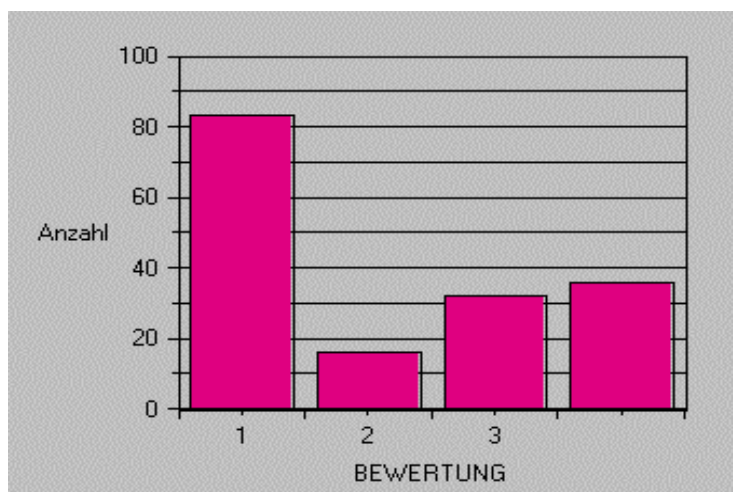


Abb. 38 : Meldungsbewertung durch den Anwender ab dem 21.5.1998

In der Darstellung wurden alle Bewertungen zum experimentellen MLM ip9 („Verträglichkeit Systemerkrankung und Metastaselokalisation“) und zum fehlerhaften MLM ip3 („Vollständigkeit Metastaselokalisation und TNM-Metastase“) vom 8.6. bis 10.6. nicht berücksichtigt.

Die Abbildung belegt, daß

- die vereinfachte Beurteilungsform deutlich häufiger verwendet wurde und
- knapp 70 % aller bewerteten Meldungen als „inhaltlich OK“ eingestuft wurden.

6.2.3. Analyse einiger vom Benutzer als „inhaltlich falsch“ eingestufte Meldungen

Da ungerechtfertigte Meldungen sehr belastend für die Benutzerakzeptanz der wissensverarbeitenden Funktionen sind, wurden einige der mit „inhaltlich falsch“ und „kann (noch) nicht beurteilt werden“ gekennzeichneten Meldungen analysiert.

Hierbei zeigte sich, daß

1.) 9 der 15 als „inhaltlich falsch“ bewerteten Meldungen zur Lokalisation-Histologie-Verträglichkeit bei vergleichbaren auf [Grundmann 1997] basierenden Prüfungen nicht generiert worden wären. Grund für das Validitätsproblem des den Prüfungen zugrundeliegenden IARC-Fachwissens aus [Parkin 1994] dürfte die mangelnde Aktualität sein. Beispiele für diese Meldungen sind :

- Nach den Empfehlungen der IARC ist bei Frau ***** die Histologie (Schlüssel: 84413 aus Familie: Ovarian tumours) mit keiner der dok. Lokalisationen (Schlüssel: 549) verträglich ! (Prüfung IARC10)

Nach [Grundmann 1997] ist die Histologie „Seröses Adenokarzinom o.n.A.“ (Schlüssel 84413^a) an Tumoren des Ovars (Schlüssel C56), des Corpus uteri (Schlüssel C54) und Cervix uteri (Schlüssel C53) möglich. Der der Meldung zugrundeliegende Sachverhalt ist konform zu diesen Angaben. Nach [Parkin 1994] ist diese Histologie nur an den Lokalisationen Ovar (Schlüssel C56.9), „andere und nicht näher bezeichnete Bereiche der weiblichen Genitalorgane“ (Schlüssel C57), „andere und mangelhaft bezeichnete Lokalisationen,, (Schlüssel C76) und „Unbekannte Primärlokalisation“ (Schlüssel C80.9) möglich.

^a alle verwendeten Histologie/ und Lokalisationscode stammen - sofern nicht anders angegeben - aus [WHO 1990]

- Nach den Empfehlungen der IARC ist bei Frau ***** die Lokalisation (Schlüssel: 530) mit KEINER der dok. Histologien aus der/n Familie/n Endometrioid carcinoma verträglich ! (Prüfung IARC13 Fall 2) (Hist.-Schlüssel : 83803)

Nach [Grundmann 1997] ist die Histologie „Endometrioides Karzinom“ (Schlüssel 83803) an Tumoren des Ovars (Schlüssel C56), des Corpus uteri (Schlüssel C54), des Cervix uteri (Schlüssel C53) und der Vagina (Schlüssel C52) möglich. Der der Meldung zugrundeliegende Sachverhalt ist konform zu diesen Angaben. Nach [Parkin 1994] ist diese Histologie nur an den Lokalisationen Corpus uteri (Schlüssel C54), Ovar (Schlüssel C56.9), „andere und nicht näher bezeichnete Bereiche der weiblichen Genitalorgane“ (Schlüssel C57), „andere und mangelhaft bezeichnete Lokalisationen,, (Schlüssel C76) und „Unbekannte Primärlokalisation“ (Schlüssel C80.9) möglich.

2.)5 der 15 als „inhaltlich falsch“ bewerteten Meldungen zur Lokalisation-Histologie-Verträglichkeit auch bei auf [Grundmann 1997] basierenden Prüfungen generiert worden wären. Dies legt den Schluß nahe, daß auch die aktuelleren Angaben nicht optimal an die klinische Routine angepaßt sind. Beispiele für diese Meldungen sind :

- Nach den Empfehlungen der IARC ist bei Frau ***** die Lokalisation (Schlüssel: 4927) mit KEINER der dok. Histologien aus der/n Familie/n Melanomas verträglich ! (Prüfung IARC13 Fall 2) (Hist.-Schlüssel : 87422)
- Nach den Empfehlungen der IARC ist bei Herrn ***** die Histologie (Schlüssel: 95310 aus Familie: Meningeal tumours) mit keiner der dok. Lokalisationen (Schlüssel: 718) verträglich ! (Prüfung IARC10)

3.)9 der 10 mit „kann (noch) nicht beurteilt werden“ bewerteten Meldungen zur Lokalisation-Histologie-Verträglichkeit auch von auf [Grundmann 1997] basierenden Meldungen generiert worden wären.

4.)einige als „inhaltlich falsch“ eingestufte Meldungen auf Unachtsamkeiten beim Entwurf der Prüfung zurückzuführen sind. Beispiele hierzu sind :

- Nach den Empfehlungen der IARC ist bei Frau ***** die Histologie (Schlüssel: 87433 aus Familie: Melanomas) mit keiner der dok. Lokalisationen (Schlüssel: 173) verträglich ! (Prüfung IARC10)

Die Unachtsamkeit beruht darauf, daß nicht sichergestellt wurde, daß die Prüfung nur auf Sachverhalte mit Lokalisationen, die nach [WHO 1990] codiert wurden, angewendet wird. Der Lokalisationscode 173 aus [WHO 1976] ist nicht in der implementierten Liste der verträglichen Lokalisationen zur histologischen Familie „Melanomas“ enthalten.

- Bei Herrn ***** wurde ein Operationsdatum dokumentiert, welches in der Zukunft liegt. Bitte auf Flüchtigkeitsfehler kontrollieren ! (Integritätsprüfung 1)

Hier wurde bei der Realisierung der Integritätsbedingung nicht differenziert zwischen geplanten und stattgefundenen Operationen.

5.)viele medizinische Integritätsbedingungen den Charakter bewährter Daumenregeln haben.

Hierzu soll folgendes Beispiel genannt werden :

- Maligne Tumoren mit der/n Histologie/n (BURKITT-LYMPHOM (BL) (K, W) ANM.) sind bei Patienten über 14 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich. (zu Herrn ***** Histologie : 9687 Alter 30)

Der dieser Meldung zugrundeliegende dokumentierte korrekte Sachverhalt wird nach [Parkin 1994] als sehr unwahrscheinlich eingestuft. Dies führte zur Generierung dieser mit „inhaltlich falsch“ bewerteter Meldung.

Berücksichtigt man diese Ergebnisse kann man feststellen, daß aufgrund der Punkte 2.) und 3.) der Anteil gerechtfertigter Hinweise größer ist als in Abbildung 38 dargestellt. Eine weitere Reduzierung ungerechtfertigter Meldungen muß angestrebt werden, kann aber nur in einem langfristigen Erfahrungsprozeß erreicht werden. Folgende Probleme müssen dabei berücksichtigt werden :

- die oftmals heuristische Natur (obiger Punkt 5) medizinischer Integritätsbedingungen (vgl. auch die Anmerkungen zu den methodischen Einschränkungen von Integritätsprüfungen in Kapitel 5.3.2.),
- der Aufwand bei der Erstellung von allgemein akzeptierten, aktuellen und edv-verarbeitbaren Fachwissens über Lokalisation-Histologie-Verträglichkeiten (obiger Punkt 1.).

6.2.4. Zusammenfassung der ersten Ergebnisse

- Der in die Shell integrierte retract-Mechanismus erlaubt die Reduktion des Meldungsaufkommens für den Benutzer um knapp Faktor 2 durch Filterung doppelter und inhaltlich veralteter Meldungen. Das Vorkommen solcher Meldungen ist ein typisches Phänomen bei Systemen dieses Typs und ist bedingt durch die automatische Auslösung in Verbindung mit einer meist zur Meldungsgenerierung nicht synchronen Meldungspräsentation.
- Nach den bisherigen Zahlen führte eines von hundert aufgerufenen MLMs zu einem Meldungshinweis. Beim durchschnittlichen Dokumentationsaufkommen führt dies täglich zu insgesamt 3-5 Meldungen.
- Im Zeitraum bis Ende Oktober wurden knapp 250 Hinweise auf Dateninkonsistenzen und Unvollständigkeiten den Dokumentationskräften präsentiert.
- Knapp 70 % der bewerteten Meldungen wurden vom Benutzer als „inhaltlich OK“ eingestuft. Die Untersuchung aller mit „kann (noch) nicht beurteilt werden“ und mit „inhaltlich falsch“ bewerteten Meldungen zur Verträglichkeit von Lokalisationen und Histologien hat gezeigt, daß 5 der 15 mit „inhaltlich falsch“ und 9 der 10 mit „kann (noch) nicht beurteilt werden“ bewerteten Meldungen nach den Vorgaben sowohl aus [Parkin 1994], die ja Grundlage der durchgeführten Integritätsprüfung sind, als auch aus [Grundmann 1997] als korrekt einzustufen sind. Der Anteil der aufgedeckten Verstöße gegen Integritätsprüfungen dürfte deshalb noch höher sein als der Anteil vom Benutzer „als inhaltlich OK“ eingestufte Meldungen.

6.3. *Bewertung des ersten Einsatzes*

Der erste Einsatz hat gezeigt, daß mit der entworfenen Shell für wissensverarbeitenden Funktionen eine klinisch einsetzbare Unterstützung bei der dokumentationsnahen Erkennung von Dateninkonsistenzen und Unvollständigkeiten geleistet werden kann. Voraussetzungen für eine klinische Einsetzbarkeit ist die Akzeptanz *aller* beim Einsatz beteiligten Personengruppen.

- Befürchtungen seitens des *GTDS-Systemverwalters* über eine Beeinträchtigung der normalen Abläufe konnten ausgeräumt werden. In den sechs Monaten Einsatz ist keine Komplikation, die in Zusammenhang mit der Shell stehen könnte, bekannt geworden.
- Die Wartung der wissensverarbeitenden Funktionen durch den *Wissensverwalter* war weitgehend problemlos. Dies ist im wesentlichen auf die strenge Modularität der Wissensbank zurückzuführen. Notwendige Tätigkeiten des Wissensverwalters waren
 1. die Initialisierung der Shell zum Löschen der Systemtabellen und zum Export von Daten zur Auswertung der Shellaktivität,
 2. das Verfolgen der Systemaktivität einschließlich der durchgeführten Benutzerbewertungen mit dem Ziel des Aufdeckens von Fehlern in der Funktionsweise der Funktionen und
 3. das Entfernen der so aufgedeckten Fehler.

Die Verantwortlichkeit für das medizinische Fachwissen insbesondere zur Verträglichkeit von Lokalisationen und Histologien beschränkt sich auf die Auswahl vorhandenen Wissens. Das verwendete IARC-Fachwissen aus [Parkin 1994] ist inhaltlich nicht immer mit dem alternativ verwendbaren aktuellerem Fachwissen aus [Grundmann 1997] identisch. Eine Analyse von fünfzehn auf [Parkin 1994] basierenden Meldungen zur Lokalisation-Histologieunverträglichkeit, die vom Benutzer als „inhaltlich falsch“ eingestuft wurden, ergab, daß davon neun bei auf [Grundmann 1997] basierenden Meldungen nicht generiert worden wären. Ob das Wissen aus [Grundmann 1997] besser als Grundlage einer Prüfung dient, bedarf jedoch noch weiterer Untersuchung : so wären fünf der obigen fünfzehn als „inhaltlich falsch“ eingestuften Meldungen auch nach [Grundmann 1997] generiert worden. Weiterhin fielen bei [Grundmann 1997] auch einige Unvollständigkeiten in der Liste der zu einer Histologie verträglichen Lokalisationen auf. Beispielsweise fehlen bei der Histologie 83803 „Endometrioides Karzinom“ die im IARC-Fachwissen enthaltenen Lokalisationen C57 „Andere und nicht näher bezeichnete Bereiche der weiblichen Genitalorgane“, C76

„Andere und mangelhafte bezeichnete Lokalisationen“ sowie C80 „Unbekannte Primärlokalisation“.

- Wichtiger Faktor zur Erreichung der Akzeptanz durch den *Benutzer* ist neben einem minimalen zusätzlichen Arbeitsaufwand (1) eine nicht zu große Menge von Meldungen und (2) möglichst wenige ungerechtfertigte Meldungen. Die dargestellten Ergebnisse sprechen für eine positive Benutzerakzeptanz.

Aus der klinischen Verwendbarkeit im hier verwendeten Sinne ergibt sich ein Nutzen im Anwendungsbereich. Dieser liegt bei den realisierten wissensverarbeitenden Funktionen in

- einem geringeren Arbeitsaufwand - verglichen mit einer zur Dokumentation zeitlich stark versetzten Durchführung von Integritätsprüfungen - zur Erreichung einer besseren Datenqualität sowie
- einer Reduktion von auf falschen oder unvollständigen Daten beruhenden Fehlentscheidungen.

7. Ergebnisse

Ausgangspunkt der vorliegenden Arbeit war das Ziel, durch wissensverarbeitende Funktionen eine über das bisherige Maß hinausgehende Unterstützung des medizinischen Personals zu erreichen. Hierzu wurde eine an das Gießener Tumordokumentationssystem gekoppelte Shell für wissensverarbeitende Funktionen entwickelt und schwerpunktmäßig zur erfassungsnahen Unterstützung bei der Erkennung von Dokumentationsunplausibilitäten und -unvollständigkeiten verwendet.

Zentrale Rahmenbedingung bei der Konzeption und Entwicklung war die Erreichung der klinischen Einsetzbarkeit.

Das entwickelte System zeichnet sich insbesondere

- durch seine enge Kopplung an ein vorhandenes Informationssystem, wodurch die Realisierung *integrierter* wissensverarbeitender Funktionen möglich wird,
- durch seine - softwaretechnisch gesehene - große Unabhängigkeit vom zugrundeliegenden Informationssystem,
- durch eine leichte Nachvollziehbarkeit der Funktionsweise für alle bei einem Einsatz beteiligten Personengruppen,
- durch die Ermöglichung von wissensverarbeitenden Funktionen zum Entscheidungsmonitoring und
- durch die Verwendung der Arden-Syntax als externen Formalismus zur Wissensrepräsentation, der eine leichte Wartung der Wissensbank und ein relativ problemloses Knowledge Sharing erlaubt,

aus.

Neben der entwickelten Shell für wissensverarbeitende Funktionen als Hauptergebnis der Arbeit betreffen weitere Ergebnisse

- Ergänzungen und Verwendungshinweise zur Arden-Syntax,
- Anwendung der Shell zur erfassungsnahen Erkennung von Dokumentationsfehlern
- den klinischen Einsatz in einem Tumorzentrum.

7.1. Ergänzungen und Verwendungshinweise zur Arden-Syntax

7.1.1. Ergänzungen am Sprachumfang der Arden-Syntax

Um die Realisierung wissensverarbeitender Funktionen zur Sicherstellung von Integritätsbedingungen zu erleichtern, wurde die Arden-Syntax um die Funktion `substr` ergänzt. Diese Funktion dient dazu, aus einer gegebenen Zeichenkette einen spezifizierbaren Teil herauszuschneiden und erleichtert so die Verarbeitung von Codierungen, die aus hierarchischen Schlüsselsystemen stammen. So lassen sich beispielsweise bei der Integritätsprüfung zur Sicherstellung der Plausibilität von Lokalisation und Histologie nun problemlos sämtliche aus dem GTDS gelesenen Lokalisationscodes auf einen allgemeineren dreistelligen Schlüssel reduzieren. Weiterhin wurde innerhalb der MLMs die Variable `MELDUNGSMODUS` zur Verfügung gestellt, die von der Arden-Shell bei der Meldungspräsentation interpretiert wird. Hierdurch wird es möglich, Meldungen eines MLMs dynamisch als Information, Warnung oder Fehlermeldung optisch zu kennzeichnen. Diese Möglichkeit wirkt sich positiv auf die Benutzerakzeptanz aus.

7.1.2. Verwendungshinweise

Bei der Verwendung der Arden-Syntax zur Realisierung wissensverarbeitender Funktionen sollten immer möglichst spezifische Ereignisse zur Auslösung definiert werden, da sich dies in der Regel positiv auf die Laufzeitperformance und die Nachvollziehbarkeit der wissensverarbeitenden Funktion auswirkt. Zum letzten Aspekt wird in Kapitel 4.5.2. ein Beispiel aufgeführt, in dem durch eine der obigen Empfehlung entsprechende Ereignisdefinition die Anzahl der Codezeilen innerhalb des Logic-Slots von 26 auf 12 Zeilen reduziert werden konnte. Weiterhin wurde in Kapitel 4.5.5. gezeigt, daß der Rückgriff auf zentral - außerhalb eines MLMs - abgelegtes Fachwissen aus einem MLM heraus (1) ohne Modifikationen des Sprachumfanges möglich und weiterhin (2) bei einem großen Umfang des Fachwissens auch sinnvoll ist. Dies wurde an einem MLM zur Sicherstellung der Plausibilität von Lokalisations- und Histologieangaben demonstriert, bei dem das nötige Fachwissen zentral in einer relationalen Tabelle des zugrundeliegenden Informationssystems abgelegt wurde und über im Data-Slot definierte Variablen vom Typ Liste dem MLM zur weiteren Verarbeitung zur Verfügung gestellt wurde. Bei diesem Beispiel ist diese Repräsentation von Fachwissen sinnvoll, da sie - aufgrund des Wissensumfanges - (1) übersichtlicher, (2) einfacher zu pflegen ist und

zudem (3) auch in anderen Problemkontexten (Bsp. Einschränkung der Histologiewerteliste bei bereits dokumentierter Lokalisation oder für „Offline“-Integritätsprüfungen) verwendet werden kann.

7.2. Anwendung der Arden-Shell im GTDS

Mit Hilfe der entwickelten Systemumgebung wurden wissensverarbeitende Funktionen zur erfassungsnahen Erkennung von Dokumentationsfehlern definiert. Derzeit umfaßt die Wissensbasis ca. 20 MLMs, von denen jedes die Einhaltung einer Integritätsbedingung überwacht. Neben einigen für das Datenmodell des GTDS spezifischen Prüfungen wurden mit der Methode die von der International Agency for Research on Cancer (IARC) vorgeschlagenen Integritätsprüfungen für epidemiologische Krebsregister [Parkin 1994] realisiert. Diese umfassen Prüfungen zur Sicherstellung der Vollständigkeit der Datensätze, zur korrekten Sequenz zeitlicher Angaben und Konsistenz unterschiedlicher Items.

Hierbei hat sich gezeigt,

- daß die Shell die Durchführung von Integritätsprüfungen unmittelbar während der Dokumentation am GTDS erlaubt. Die Aktivität dieser „Online-Prüfungen“ besteht aus Sicht des Arztes oder des Dokumentars lediglich aus einer Meldung, die im Fehlerfall aufgeblendet wird. Aufgrund der Reaktionszeit des Systems im Sekundenbereich kann der Anwender einem Verstoß gegen eine definierte Integritätsbedingung, noch während er sich in der entsprechenden Dokumentationsmaske befindet und die entsprechenden Akten vorliegen hat, auf den Grund gehen,
- daß sich sämtliche von [Haag 1992] unterschiedenen Typen von Integritätsprüfungen realisieren lassen. Haag unterscheidet sämtliche Integritätsprüfungen danach, ob sie auf Attribut-, auf Relationen- oder auf Datenbankebene ansetzen. Insbesondere die Integritätsprüfungen auf Relationenebene erfordern bei der Systemkonzeption für Online-Ansätze zur Deadlock-Vermeidung besondere Aufmerksamkeit, da bei vorhandenen Datenbanksystemen oft Lesezugriffsrestriktionen („Mutating-Table-Restriktion“) auf die einen Trigger auslösenden Relation bestehen,
- daß bei Integritätsprüfungen, denen größere Mengen Fachwissen zugrundeliegen, von einer Formulierung dieses Fachwissens innerhalb des prozeduralen Codes des Logic-Slots abzusehen ist. Dies ist der Fall bei Prüfungen zur Verträglichkeit von Histologie und

Lokalisation. Stattdessen sollte dieses Fachwissen zentral in deklarativer Form abgelegt werden (vgl. Kapitel 7.1.).

Als weiteres Anwendungsgebiet wurde eine Studienzuteilungsunterstützung untersucht. Hier wurde anhand eines Beispiels gezeigt, daß es mit der Arden-Syntax möglich ist, eine relativ differenzierte Einschätzung über die Eignung eines Patienten zur Teilnahme an einer Studie zu machen. Durch die Entscheidung genau eine Studienentscheidung in einem MLM abzulegen und die Wiederverwendung des entworfenen Scoring-Algorithmus, kann zudem eine flexible Anpassung an sich ändernde Studienbedingungen erreicht werden.

7.3. *Klinischer Einsatz in einem Tumorzentrum*

Seit Mai 1998 wird die Shell für wissensverarbeitende Funktionen mit den genannten MLMs im Tumorzentrum Cottbus eingesetzt. Im Zeitraum bis Ende Oktober wurden dabei knapp 250 Hinweise auf Dateninkonsistenzen und Unvollständigkeiten den vier Dokumentationskräften präsentiert. Knapp 70% der bewerteten Meldungen wurden als „inhaltlich OK“ eingestuft. Der klinische Einsatz hat weiterhin das massive Auftreten störender Meldungen, also Meldungen, die schon zum Präsentationszeitpunkt nicht mehr aktuell oder mit einer anderen Meldung identisch sind, und damit die Bedeutung des in die Shell eingebauten retract-Mechanismus zur Reduktion ungerechtfertigter Meldungen gezeigt. Im Schnitt wurde knapp die Hälfte aller generierten Meldungen für den Benutzer ausgeblendet. Die Integration der Funktionen, die überschaubare Menge von Meldungen sowie der geringe Anteil ungerechtfertigter Meldungen sind wichtige Faktoren für eine gute Benutzerakzeptanz. Die positiven Erfahrungen des GTDS-Systemverwalters bzgl. unerwünschter Interaktionen mit dem GTDS sowie des Wissensverwalters hinsichtlich der unproblematischen Pflege der Wissensbank untermauern die klinische Verwendbarkeit der realisierten wissensverarbeitenden Funktionen.

Seit Dezember 1998 wird die Shell ebenfalls im Tumorzentrum Frankfurt/Oder eingesetzt. Ein Einsatz bei weiteren Anwendern ist geplant.

8. Diskussion

8.1. *Methodische Anforderungen zur Erreichung einer klinischen Einsetzbarkeit wissensverarbeitender Funktionen*

Die Erreichung einer klinischen Einsetzbarkeit war das zentrale Entwurfsziel der vorliegenden Arbeit. Neben Anforderungen, wie der Wahl eines sinnvollen Anwendungsgebietes und einer entsprechenden Motivation der Entwickler, wurden methodische Anforderungen an wissensverarbeitende Funktionen herausgearbeitet, um eine klinische Einsetzbarkeit zu gewährleisten. Diese methodischen Anforderungen sind in Abbildung 36 dargestellt und werden anschließend diskutiert.

Leichte Nachvollziehbarkeit der Funktionsweise
Problemlose Wartbarkeit der Wissensbasis
Hohe Integration in zugrundeliegendes Informationssystem
Katalytische Unterstützung der Entscheidungsprozesse
Knowledge Sharing
Eingeschränktes Knowledge Reusing
„retract“-Mechanismus

Abb. 36 : Methodische Anforderungen zur Erreichung einer klinischen Einsetzbarkeit

Die *leichte Nachvollziehbarkeit* der Funktionsweise des Systems ist wichtig für die Akzeptanz durch den Benutzer. Abgesehen von psychologischen Gründen ist sie Voraussetzung dafür, daß der Benutzer Modifikationen an vorhandenen MLMs vorschlagen oder gar neue MLMs anregen kann, wodurch eine Anpassung an spezielle Unterstützungswünsche möglich wird. Sie wird zum einen dadurch erreicht, daß eine Systemaktivität auf ein einziges MLM zurückzuführen ist, und zum anderen dadurch, daß der Ableitungsvorgang im Logic-Slot in prozeduraler Form beschrieben ist. Die Einschätzung, daß die prozedurale (und nicht eine alternative deklarative) Darstellung auch von Nicht-EDV-Experten besser verstanden werden kann, wird auch von anderen Autoren [Hripcsak 1994] beschrieben.

Eine relativ *problemlose Wartung der wissensverarbeitenden Funktionen* ist elementar für die Akzeptanz durch den Wissensverwalter. Sie wurde erreicht durch die Strukturierung der Wissensbasis in voneinander unabhängige Wissensmodule (MLMs), wobei jedes Modul das gesamte nötige Wissen zur Durchführung einer integrierten wissensverarbeiteten Funktion

enthält. Durch Löschen, Hinzufügen oder Modifizieren von Funktionen können andere Funktionen nicht in ihrer Funktionsweise beeinträchtigt werden. Weiterhin müssen bei der Verifikation einer Funktion keine unüberschaubaren Abhängigkeiten zu anderen Modulen berücksichtigt werden. Die Beschränkung der Verwendungsmöglichkeit eines Wissenselementes auf eine wissensverarbeitende Funktion ist eine einfache Form des „role-limiting“, wie es bei vielen Ansätzen der modellbasierten Wissensmodellierung verwendet wird [Musen 1995].

Eine *hohe Integration* in ein zugrundeliegendes Informationssystem führt dazu, daß die Benutzer die Vorteile des Systems erfahren, ohne selbst Arbeitsaufwand einbringen oder ihre normalen Arbeitsabläufe modifizieren zu müssen. Sie wurde dadurch erreicht, daß die Aktivitäten des Systems automatisch nach Auftreten zuvor definierter Ereignisse ausgelöst werden, die dafür benötigten Fakten aus dem Informationssystem gelesen werden und generierte Meldungen unmittelbar präsentiert werden. Die Integriertheit wissensverarbeitender Funktionen ist eine wichtige Voraussetzung für eine Akzeptanz durch den Benutzer [Brigl 1998], [Mann 1995], [Spitzer 1994].

Die von [Miller 1991] vorgeschlagene Einteilung wissensbasierter Systeme in katalytische und monolithische Systeme unterscheidet diese nach der Rolle, die dem Benutzer bei der Systembenutzung zugedacht wird. Eine *katalytische Unterstützung* der menschlichen Entscheidungsprozesse ist dadurch gekennzeichnet, daß der Benutzer

1. entsprechend seines individuellen Stils,
2. unter Verwendung seines Fachwissens,
3. seiner zum Teil unbewußten oder schwer operationalisierbaren Kenntnisse zum konkreten Fall und
4. seiner persönlichen Erfahrung und Intuition

seine Entscheidungsprozesse durchführen kann. Lediglich bei Übersehen wichtiger Tatsachen oder Nichtberücksichtigung eventueller neuer Erkenntnisse wird er durch das System unterstützt. [Miller 1991] weist daraufhin, daß nur „katalytisch“ ausgelegte wissensverarbeitende Funktionen eine Chance auf Akzeptanz beim Benutzer haben. Ein weiterer wichtiger Effekt dieser Art der Unterstützung auf die Realisierbarkeit wissensverarbeitender Funktionen liegt darin,

1. daß diese nur Wissen über Bruchstücke des Entscheidungsvorganges - typischerweise da, wo der Benutzer erfahrungsgemäß oft Fehler macht - beinhalten müssen. Hierdurch wird das Problem der Vollständigkeit der Wissensakquisition entschärft.
2. daß eine Erklärungskomponente nur selten notwendig ist, da der Grund für die Entscheidungsunterstützung weniger in der Unkenntnis als im *Übersehen* bestimmter Sachverhalte liegt. Die Bedeutung des Nicht-Notwendigseins einer Erklärungskomponente ist deshalb besonders hoch einzuschätzen, da die Realisierung einer vom Benutzer akzeptierten Erklärungsfunktion als problematisch einzustufen ist [Miller 1991].

Ein erklärtes Designziel der Arden-Syntax ist die Ermöglichung des Austausches von Wissen (*Knowledge Sharing*) zwischen unterschiedlichen Institutionen, um so den Flaschenhals der Wissensakquisition so überwinden. Sie wird - wie auch die leichte Nachvollziehbarkeit der Funktionsweise und die leichte Wartbarkeit der Wissensbasis - durch die Strukturierung der Wissensbank in voneinander unabhängige MLMs. Hierdurch ist sichergestellt, daß durch ein Hinzufügen eines MLMs in eine Wissensbank aus MLMs keine unerwünschten Seiteneffekte zu befürchten sind. Dies ist ein typisches Problem bei Systemen, in denen im Rahmen einer Problemlösung eine Inferenzmaschine auf unterschiedlichste Bestandteile einer Wissensbasis zugreifen kann. Praktische Untersuchungen [Pryor 1994] haben jedoch gezeigt, daß ein Austausch von MLMs zu Anpassungen eines MLMs an lokale Gegebenheiten führen kann. Diese sind im wesentlichen auf unterschiedliche Schnittstellen zum Datenbestand der entsprechenden Informationssysteme zurückzuführen. Das Problem der durch unterschiedliche Schnittstellen bedingten Anpassungen beim Knowledge Sharing darf jedoch nicht überbewertet werden. Zum einen, weil mitunter unterschiedliche Institutionen die gleichen Informationssysteme (mit identischer Schnittstelle) haben und so untereinander problemlos MLMs austauschen können, zum anderen, weil durch einen guten Programmierstil eine gute Nachvollziehbarkeit der Funktionsweise einer Problemlösung erreicht werden kann, durch die eventuell nötige Anpassungen machbar werden sollten. Der hier geforderte Programmierstil umfaßt mehrere Aspekte, wie z. B.

- die Verwendung möglichst spezifischer auslösender Ereignisse. Hierdurch kann der Logic-Slot - wie in Kapitel 4.5.2. gezeigt - von überflüssigem Ballast befreit werden.
- die Realisierung einer wissensverarbeitenden Funktion durch genau ein MLM. Dies bedeutet einen Verzicht auf die in der Arden-Syntax vorgesehene Möglichkeit, ein MLM aus einem anderen aufzurufen.

Ein *Knowledge Reusing* von umfangreichem Fachwissen ist in manchen Fällen für die Akzeptanz durch den Wissensverwalter unabdingbar. Dies hat sich in Kapitel 5.3.3.14. bei den realisierten MLMs zur Verträglichkeitskontrolle von Histologien und Lokalisationen gemäß [Parkin 1994] gezeigt. Hier wäre eine redundante Haltung des für mehrere MLMs identischen umfangreichen Fachwissens nicht zumutbar gewesen. Das Knowledge Reusing wurde durch Ablage des in [Parkin 1994] formulierten Wissens in relationalen Tabellen erreicht, auf die aus den MLMs über den Data-Slot zugegriffen wurde. Hierzu waren keine Sprachmodifikationen an der implementierten Arden-Syntax nötig. Weiterer Vorteil ist die erleichterte Wartung dieses Fachwissens. Einen ähnlichen Ansatz zur Wiederverwendung medizinischen Fachwissens zeigen [Linnarson 1989] und [Prokosch 1991], die vorschlagen, das medizinische Fachwissen in einem semantischen Netz zu modellieren und den Wissensmodulen als Infrastruktur zur Verfügung zu stellen. Ein solcher Ansatz erlaubt zwar ein auf Faktenwissen beschränktes Knowledge Reusing, erschwert jedoch Knowledge Sharing, da neben einzelnen MLMs nun noch das separat repräsentierte Fachwissen weitergegeben werden muß.

Der „*retract*“-*Mechanismus* dient der Ausblendung von Meldungen, die zum Zeitpunkt der Präsentation bereits veraltet oder identisch mit einer anderen sind. Ihr Auftreten ist ein typisches Phänomen bei integrierten wissensverarbeitenden Funktionen. [Kuperman 1997] berichtet von störenden Meldungen im Brigham Integrated Computing System (BICS). Sie resultieren aus der automatischen Auslösung der Funktionen verbunden mit einer meist nicht synchronen Meldungspräsentation und haben nichts mit Fehlern in der Wissensbank zu tun. Dieser Mechanismus ist ein wichtiger Faktor zur Erreichung einer guten Benutzerakzeptanz. Der realisierte „*retract*“-*Mechanismus* gilt allgemein für alle wissensverarbeitenden Funktionen und ist transparent für den MLM-Entwickler.

8.2. *Klinischer Einsatz*

Der bisherige Einsatz in den Tumorzentren Cottbus und Frankfurt/Oder hat die klinische Einsetzbarkeit der Shell für integrierte wissensverarbeitende Funktionen gezeigt. Dies ist auf den grundlegenden Designansatz der Berücksichtigung *aller* bei einem klinischen Einsatz involvierten Personengruppen zurückzuführen. Wesentliche Folgen dieses Ansatzes sind

- die Entfernung störender Meldungen. Diese resultieren aus einer automatischen Auslösung der Funktionen verbunden mit einer nicht asynchronen Meldungspräsentation, wodurch beispielsweise eine Meldung zum Präsentationszeitpunkt schon veraltet sein kann. In [Kuperman 1997] werden vergleichbare störende Meldungen im Rahmen eines „Alerting“-Systems innerhalb des Brigham Integrated Computing System (BICS) beschrieben. Die Ergebnisse des bisher halbjährigen Einsatzes im Tumorzentrum Cottbus zeigen, daß hierdurch das Meldungsaufkommen für den Benutzer um Faktor 3 reduziert wurde. Den Benutzern wurden so insgesamt knapp 250 Meldungen präsentiert. Knapp 70 % der bewerteten Meldungen wurde als „inhaltlich OK“ eingestuft.
- die SW-mäßige Unabhängigkeit vom vorhandenen Informationssystem. Neben der Akzeptanz durch den Systemverwalter des zugrundeliegenden Informationssystem, der verständlicherweise jedem Eingriff am Informationssystem skeptisch gegenübersteht, wurde ein Einsatz bei allen den Datenbestand bearbeitenden Anwendungen möglich. Hierzu gehören sowohl die neuentwickelte graphische Version des vorhandenen Informationssystems als auch vorhandene Schnittstellen zur Datenkommunikation (BDT, HL7).
- die Verwendung der Arden-Syntax als Formalismus zur Wissensrepräsentation. Hier ist vor allem der positive Einfluß der Kapselung des Wissens einer integrierten wissensverarbeitenden Funktion in ein unabhängiges Modul auf die Kontrolle und Wartbarkeit durch den Wissensverwalter zu nennen. Während des Einsatzes durchgeführte Modifikationen an der modularen Wissensbank (Modifikation, Löschen und Hinzufügen von Funktionen) waren problemlos für den Wissensverwalter.

Als erstes Anwendungsfeld wurde die erfassungsnahe Unterstützung des dokumentierenden Personals bei der Erkennung von Dateninkonsistenzen und Datenunvollständigkeiten gewählt. Hierbei wird automatisch nach einer Dateneingabe die Einhaltung bestimmter Integritätsbedingungen kontrolliert und gegebenenfalls ein Hinweistext für den Benutzer generiert. Der besondere Nutzen dieses Ansatzes im Vergleich zu alternativen typischerweise zeitlich stark zur Erfassung versetzten Prüfverfahren liegt im geringeren Aufwand bei der Fehlerbereinigung,

da dem erfassenden Personal der Fall noch gedanklich präsent ist und die zugrundeliegenden Dokumente (Krankenakten, Erfassungsbögen) noch greifbar sind. [Van Holten 1991] verweist auf eine wachsende Popularität solcher Ansätze. Ein sinnvoller Ausbau dieses Ansatzes

- ist die Kennzeichnung eines überprüften Datensatzes mit einer diesbezüglichen Angabe. Eine solche Angabe existiert als sogenanntes „override flag“ innerhalb des „United States Surveillance, Epidemiology and End Results Program“ (SEER) [Van Holten 1991]. Hierdurch können vor allem bei sehr seltenen Sachverhalten Nachfragen übergeordneter Registerstellen vermieden werden.
- ist die Zusammenfassung von generierten Hinweisen zu Fehlerberichten und deren Bereitstellung als Arbeitsgrundlage für periodisch stattfindende Datenkontrollen durch Prüfarzte.

Weitere denkbare Anwendungsfelder im Umfeld Klinischer Krebsregister sind die Studienzuteilung von Patienten, der Hinweis auf Alarmsituation oder die Verbreitung klinischer Leitlinien. Im Einzelfall muß entschieden, ob solche Funktionen sinnvoll sind oder nicht. Die in Kapitel 5.1. formulierten Anforderung stellen hier eine Entscheidungshilfe dar. Eventuell kann durch sinnvolle wissensverarbeitende Funktionen eine Akzeptanzsteigerung klinisch behandlungsnaher Dokumentation im Rahmen von Arztarbeitsplätzen erreicht werden.

9. Zusammenfassung

Das Ziel dieser Arbeit war es, durch wissensverarbeitende Funktionen eine über das bisherige Maß hinausgehende Entscheidungsunterstützung für Benutzer im Umfeld Klinischer Krebsregister zu erreichen. Hierzu wurde das Gießener Tumordokumentationssystem (GTDS) um eine als Shell für wissensverarbeitende Funktionen bezeichnete Systemumgebung erweitert, die den Aufbau einer Wissensbank aus Medical Logic Modules (MLMs) erlaubt und die Operationalität ihrer Elemente von der Ereigniserkennung bis zur Meldungspräsentation sicherstellt. Besondere Kennzeichen dieser Shell sind

1. bedingt durch die Unabhängigkeit der MLMs eine leichte Validierung einzelner wissensverarbeitender Funktionen, ein relativ leichter Austausch einzelner MLMs zwischen unterschiedlichen Systemen und eine leichte Pflege der Wissensbank,
2. eine Integration der realisierten wissensverarbeitenden Funktionen in dem Sinne, daß
 - sie automatisch nach bestimmten Ereignissen innerhalb des GTDS ausgelöst werden,
 - sie selbständig auf für den Ableitungsvorgang notwendige im GTDS vorhandene Patientendaten zugreifen und
 - daß eventuelle Meldungen umgehend dem GTDS-Benutzer präsentiert werden.
3. ein eingebauter Mechanismus (retract-Mechanismus) zur Ausblendung veralteter und doppelter Meldungen. Das Vorkommen solcher Meldung ist ein typisches Phänomen bei Systemen dieses Typs und ist bedingt durch die automatische Auslösung in Verbindung mit einer meist zur Meldungsgenerierung nicht synchronen Meldungspräsentation.
4. eine softwareseitige Trennung vom GTDS. Die einzige Veränderung am GTDS besteht im periodischen Aufruf einer Maske zur Präsentation ungelesener Meldungen.

Die genannten Kennzeichen stellen wichtige Faktoren zur Erreichung einer Akzeptanz der Shell für wissensverarbeitende Funktionen beim Wissensverwalter (Kennzeichen 1), beim GTDS-Anwender (Kennzeichen 2 und 3), sowie beim GTDS-Systemverwalter (Kennzeichen 4) dar. Die Akzeptanz aller beim Einsatz involvierten Personengruppen ist eine wichtige Bedingung für die klinische Einsetzbarkeit des Systems.

Da das GTDS derzeit in erster Linie von Dokumentationsassistenten bedient wird und eine verlässliche Datenqualität eine wichtige Voraussetzung für die Verwendbarkeit der Daten ist, wurden als erste Anwendung wissensverarbeitende Funktionen zur erfassungsnahen Unterstützung bei der Erkennung von Dokumentationsunvollständigkeiten und Inkonsistenzen definiert. Prinzipiell ist die Shell nicht auf ein spezifisches Anwendungsgebiet zugeschnitten. Im

Umfeld eines Arztarbeitsplatzes wären beispielsweise Funktionen zum Hinweis auf medizinische Alarmsituationen oder Behandlungsrichtlinien denkbar.

Derzeit umfaßt die Wissensbasis ca. 20 MLMs, von denen jedes die Einhaltung einer Integritätsbedingung überwacht. Neben einigen für das Datenmodell des GTDS spezifischen Prüfungen wurden mit der Methode die vom International Agency for Research on Cancer (IARC) vorgeschlagenen Integritätsprüfungen für epidemiologische Krebsregister [Parkin 1994] realisiert. Diese umfassen Prüfungen zur Sicherstellung der Vollständigkeit der Datensätze, zur korrekten Sequenz zeitlicher Angaben und Konsistenz unterschiedlicher Items (z.B. - besonders interessant wegen des Umfanges des nötigen Fachwissens - zur Vereinbarkeit von Lokalisations- und Histologieangaben).

Seit Mai 1998 wird die Shell für wissensverarbeitende Funktionen mit den genannten MLMs im Tumorzentrum Cottbus eingesetzt. Im Zeitraum bis Ende Oktober wurden dabei knapp 250 Hinweise auf Dateninkonsistenzen und Unvollständigkeiten den vier Dokumentationskräften präsentiert. Knapp 70% der bewerteten Meldungen wurden als „inhaltlich OK“ eingestuft. Der klinische Einsatz hat weiterhin das massive Auftreten störender Meldungen, also Meldungen, die schon zum Präsentationszeitpunkt nicht mehr aktuell oder mit einer anderen Meldung identisch sind, und damit die Bedeutung des in die Shell eingebauten retract-Mechanismus zur Reduktion ungerechtfertigter Meldungen gezeigt. Der implementierte retract-Mechanismus gilt allgemein für alle MLMs und ist transparent für den die MLMs entwerfenden Wissensverwalter. Die Integration der Funktionen, die überschaubare Menge von Meldungen sowie der geringe Anteil ungerechtfertigter Meldungen sind wichtige Faktoren für eine gute Benutzerakzeptanz. Die positiven Erfahrungen des GTDS-Systemverwalters bzgl. unerwünschter Interaktionen mit dem GTDS sowie des Wissensverwalters hinsichtlich der unproblematischen Pflege der Wissensbank untermauern die klinische Verwendbarkeit der realisierten wissensverarbeitenden Funktionen.

Anhang A : Glossar zentraler Begriffe

Arden-Syntax

Formalismus zur Wissensrepräsentation, der als Grundelemente die Formulierung von sogenannten Medical Logic Modules (MLM) erlaubt. Jedes MLM entspricht im Normalfall einer konkreten integrierten wissensverarbeitenden Funktion.

Shell

Als Shell wird hier eine Softwareumgebung bezeichnet, welche die Realisierung wissensverarbeitender Funktionen ermöglicht, aber selber noch kein diese Funktionen definierendes Wissen beinhaltet. Das Füllen der Shell mit Wissen entspricht dem Aufbau einer Wissensbank aus Wissensmodulen.

Integrierte wissensverarbeitende Funktionen

Wissensverarbeitende Funktionen, die automatisch aufgrund von Ereignissen in einem Informationssystem ausgelöst werden (Ablaufintegriertheit) und bei der Ableitung auf die im Informationssystem vorhandenen Daten zugreifen (Datenintegriertheit).

Störende Meldungen

Eine Meldung ist störend, wenn sie zum Zeitpunkt der Präsentation nicht mit dem aktuellen Dokumentationsstand übereinstimmt („veraltete“ Meldung) oder sie mit einer anderen präsentierten Meldung inhaltlich identisch ist („doppelte“ Meldung). Ihr Auftreten ist ein typisches Phänomen bei integrierten wissensverarbeitenden Funktionen. Sie resultieren aus der automatischen Auslösung der Funktionen verbunden mit einer meist nicht synchronen Meldungspräsentation. Die störenden Meldungen haben nichts mit Fehlern in der Wissensbank zu tun.

retract-Mechanismus

Ein Mechanismus, der bei der Anwendung integrierter wissensverarbeitender Funktionen dafür sorgt, daß der Pool der ungelesenen Meldungen weder doppelte noch inhaltlich veraltete Meldungen enthält. Bei der entwickelten Shell für wissensverarbeitende Funktionen ist dieser Mechanismus transparent für die wissensverarbeitenden Funktionen in der Arden-Engine enthalten.

Externer Formalismus zur Wissensrepräsentation

Der Formalismus, der für den Wissensverwalter bei der Entwicklung und Wartung wissensverarbeitender Funktionen maßgeblich ist. Er ist idealerweise standardisiert, kompakt und leicht verständlich. Bei der Shell für wissensverarbeitenden Funktionen ist dies die Arden-Syntax.

Interner Formalismus zur Wissensrepräsentation

Ein externe Formalismus zur Wissensrepräsentation ist in der Regel nicht unmittelbar in der vorhandenen Softwareumgebung ablauffähig. Um eine Operationalität zu erreichen wird er deshalb in einen internen Formalismus umgewandelt, der letztlich eine Sprache darstellt, die verarbeitet werden kann. Bei der Shell für wissensverarbeitende Funktionen erfolgt diese Umwandlung durch den Arden-Compiler. Der interne Formalismus zur Wissensrepräsentation besteht hier im wesentlichen aus Datenbanktriggern und Datenbankprozeduren.

Anhang B : Literaturverzeichnis

[Aho 1990]

Aho A., Sethi R., Ullman J.
Compilerbau I
Addison-Wesley 1990

[Alvey 1987]

Alvey P.L., Myers C.D. et al.
High Performance for Expert Systems - Escaping from the demonstrator class
Medical Informatics 12 (1987) S.97

[American College of Physicians 1994]

American College of Physicians and Infectious Diseases Society of America
Human immunodeficiency virus (HIV) infection
Ann Intern Med 120 (1994) S310

[Anschütz 1987]

Anschütz F.
Ärztliches Handeln: Grundlagen, Möglichkeiten, Grenzen, Widersprüche
Wissenschaftliche Buchgesellschaft Darmstadt 1987

[ASTM 1993]

ASTM Standard Specification for Defining and Sharing Modular Health Knowledge
Bases (Arden-Syntax for Medical Logical Modules)
Annual Book of ASTM-Standards, Volume 14.01, ASTM,
Philadelphia 1993 S547

[Avron1982]

Barr A., Feigenbaum E.A.
The Handbook of Artificial Intelligence - Volume 2
William Kaufmann, Inc. 1982 S182

[Bang 1997]

Bang M., Eriksson H.
Generation of Development Environments for the Arden Syntax
Proceedings of the 21th AMIA Symposium 1997

[Blois 1980]

Blois M.S.
Clinical Judgement and Computers
N Engl J Med 303 (1980) S192

[Brigl 1998]

Brigl B., Mendoza G., Ringleb P., Hacke W., Haux R.
Evaluation eines Daten- und Wissensbanksystems für die Neurologie ...
Informatik, Biometrie und Epidemiologie in Medizin und Biologie 29
(ISSN 0943-5581)
Verlag Eugen Ulmer GmbH & Co. Stuttgart

[Clancey 1984]

Clancey J.W.
Use of MYCIN's Rules for Tutoring
Buchanan B.G., Shortliffe E.H. (Eds.) : Rule-Based Expert Systems
Addison Wesley Publishing Company 1984

[Davis 1984]

Davis R., Buchanan B.G.
Meta-Level Knowledge
Buchanan B.G., Shortliffe E.H. (Eds.) : Rule-Based Expert Systems
Addison Wesley Publishing Company 1984

[Dudeck 1994]

Dudeck, J., Wächter, W.
Berichte über die Neuauflage der Basisdokumentation für Tumorkranke
Entwicklungstendenzen der Tumordokumentation in Klinik und Nachsorge
Verlag der Ferber'schen Universitätsbuchhandlung, Gießen 1994

[Dudeck 1996]

Dudeck, J.
Grußwort zur 9.Informationstagung Tumordokumentation im Rahmen des 22.
Deutschen Krebskongresses
Dudeck J., Altmann U., Dalbert U., Wächter W. (Hrsg.) : Grundlagen
qualitätssichernder Tumordokumentation
Verlag der Ferber'schen Universitätsbuchhandlung, Gießen 1997

[Froese 1994]

Froese J., Moazzami M., Rautenstrauch C., Welter H.
Effiziente Systementwicklung mit Oracle7
Addison Wesley Publishing Company 1994

[Gao 1992]

Gao X., Shahsavar N., Arkad K et al.
Design and Functions of Medical Knowledge Editors for the Arden Syntax
Proceedings on the 7th world congress on Medical Informatics (MEDINFO92)
Amsterdam: North Holland 1992 S472

[Gao 1993]

Gao X.
Realizing Medical Decision Support Systems using the Arden-Syntax as Knowledge
Representation
Doctoral Dissertation November 1993
Departement of Biomedical Engineering, Medical Informatics, Linköping University

[Goldberg 1970]

Goldberg L.R.

Man versus Model of Man: A Rationale plus some Evidence for a Method of
Improving on Clinical Inferences

Psychol. Bull. 73 1970 S422

[Goodyear 1990]

Goodyear O.M., Hobsley M. et al.

Expert systems in the medical data environment: an example from breast surgery

Proceedings of the 6th Annual Meeting on Expert Systems in Medicine

London: British Medical Informatics Society 1990 S1

[Gottlob 1990 a]

Gottlob G., Frühwirth T., Horn W.

Expertensysteme

Springer-Verlag Wien New York 1990 S7

[Gottlob 1990 b]

Gottlob G., Frühwirth T., Horn W.

Expertensysteme

Springer-Verlag Wien New York 1990 S161

[Gottlob 1990 c]

Gottlob G., Frühwirth T., Horn W.

Expertensysteme

Springer-Verlag Wien New York 1990 S63

[Grundmann 1997]

Grundmann E., Hermanek P., Wagner G.

Tumorhistologieschlüssel (2.Auflage)

Wagner G., Dudeck J., Grundmann E., Hermanek P. (Hrsg.) : Tumordokumentation in
Klinik und Praxis

Springer-Verlag Berlin Heidelberg New York 1997

[Haag 1992]

Haag U., Haux R., Kieser M.

Statistische Auswertungssysteme: eine Einführung in ihre Anwendung, Konstruktion
und Bewertung

Gustav Fischer Verlag Stuttgart 1992

[Haux 1987]

Haux R.

Expertensystem in der Medizin

Habilitationsschrift

RWTH Aachen, 1987

- [Haux 1990]
 Haux R., Mann G.
 Wissensbasierte Systeme in der Medizin und ihre Integration in Informationssysteme
 Proceedings 35. Jahrestagung der GMDS
 Berlin Springer 1991 S345
- [Heathfield 1993]
 Heathfield H.A., Wyatt J.
 Philosophies for the Design and Development of Clinical Decision-Support Systems
 Meth Inform Med 32 1993 S1
- [Hein 1991]
 Hein M., Herrmann H.H., Unbescheid G.
 Das Oracle Handbuch
 Addison Wesley Publishing Company 1991
- [Hilsenbeck 1990]
 Hilsenbeck S.G.
 Quality control practices in centralized tumor registries in North America
 J. Clin. Epidemiol. 43 (1990) S1201
- [Hripcsak 1992]
 Hripcsak G., Cimino J.J., Johnson S.B., Clayton P.D.
 The Columbia-Presbyterian Medical Center Decision-Support-System as a Model for
 Implementing the Arden-Syntax
 Proceedings of the 16th AMIA Symposium 1992
- [Hripcsak 1994]
 Hripcsak G., Ludemann P., Pryor T.A. et al.
 Rationale for the Arden Syntax
 Computers and Biomedical Research 27 (1994) S291
- [Hripcsak 1995]
 Hripcsak G.
 Transferring Medical Knowledge Bases between Different HIS Environments
 Hospital Information Systems
 H.U.Prokosch, J.Dudeck (Editors) : Hospital Information Systems
 Elsevier Science B.V. 1995
- [Jackson 1987]
 Jackson P.
 Expertensysteme - Eine Einführung
 Addison Wesley Publishing Company, Bonn 1987 S100
- [Katz 1994]
 Katz,F.R. ,Altmann,U., Müller,J., Dudeck,J.
 Funktionsumfang des Gießener Tumordokumentationssystems (GTDS)
 Tagungsband zur 8. Informationsveranstaltung Tumordokumentation
 Verlag der Ferber'schen Universitätsbuchhandlung, Gießen 1994

[Kuperman 1997]

Kuperman GJ, Hiltz FL, Teich JM
Advanced Alerting Features: Displaying New Relevant Data and Retracting Alerts
Proc AMIA 1997

[Ledley 1959]

Ledley R., Lusted L.
Reasoning foundations of medical diagnosis
Science 130 (1959) S9

[Levine 1994]

Levine J.R., Mason T., Brown D.
lex & yacc
O'Reilly & Associates, Inc 1994

[Linnarson 1989]

Linnarsson R., Wigertz O.
The Data Dictionary - A controlled vocabulary for integrating clinical databases and
medical knowledge
Methods of Information in Medicine 28 (1989) S78

[Mann 1995]

Mann G.
An Object-Oriented Model for the Integration of Knowledge-Based-Systems
Proceedings MEDINFO 95
Amsterdam: North Holland 1995 S938

[Mc Donald 1976 b]

Mc Donald C.J.
Protocol-based computer reminders: the quality of care and the nonperfectibility of man
N Engl J Med 295 (1976) S1351

[Mc Donald 1984]

Mc Donald C.J., Hui S.L. et al.
Reminders to physicians from an introspective computer medical record: a two years
randomized trial
Ann Intern Med 100 (1984) S130

[Mc Donald 1976 a]

Mc Donald C.J.
Use of computer to detect and respond to clinical events: its effect on clinical behavior
Ann Intern Med 84 (1976) S162

[Mc Donald 1992]

Mc Donald C.J., Tierney W.M., et al
The Regenstrief Medical Record System: 20 years of experience in hospitals, clinics,
and neighborhood health centers
M.D.Computing Vol 9 No 4 (1992) S206

[Miller 1990]

Miller R.A., Masarie F.E.
The Demise of the „Greek Oracle“ Model for Medical Diagnostic Systems
Meth. Inform. Med. 29 (1990) S1

[Miller 1982]

Miller R.A., Pople H.E., Myers J.D.
Internist-1, An experimental Computer-Based Diagnostic Consultant for General
Internal Medicine
New England Journal of Medicine Vol. 307 No. 3 (1982)

[Miller 1986]

Miller R.A., Masarie F.E., Myers J.D.
„Quick Medical Reference“ for Diagnostic Assistance
MD Computing Vol 3 No 5 (1996) S34

[Musen 1995]

Musen, M.A., Schreiber A.Th.
Architectures for intelligent systems based on reusable components
Artificial Intelligence in Medicine 7 (1995)
Elsevier Science B.V. 1995 S189-199

[Parkin 1994]

Parkin DM, Chen VW, et al.
Comparability and Quality Control in Cancer Registration
IARC Technical Report No. 19
IARC Press Lyon 1994

[Prokosch 1994]

Prokosch H.U.
Ein Referenzmodell für den Einsatz wissensbasierter und wissensverarbeitender
Funktionen innerhalb eines Krankenhaus-Informationssystems
Habilitationsschrift des Fachbereichs Humanmedizin der Justus-Liebig-Universität
Gießen 1994

[Prokosch 1993]

Prokosch H.U., Willems C., Krause D., Dudeck J.
Die Arden-Syntax als Standard zur Wissensrepräsentation in der Medizin: Ziele,
Anwendungen, Beispiele und Probleme
Proceedings GMDS 1993 S168

[Prokosch 1991]

Prokosch H.U., Kamm S., Wieczorek D., Dudeck J.
Knowledge Representation in Pharmacology - A possible Application Area for the
Arden-Syntax
Proceedings Annual Symposium on Computer Applications in Medical Care 1991 S243

- [Pryor 1991]
 Pryor T.A.
 Medical Knowledge Representation
 ASTM Standardization News Aug. 1991
- [Pryor 1983]
 Pryor T.A.
 The HELP System
 Journal of Medical Systems Vol. 7 No. 2 (1983) S87
- [Pryor 1994]
 Pryor TA, Hripcsak G.
 Sharing MLMS: an experiment between Columbia-Presbyterian and LDS Hospital
 Proceedings Annual Symposium on Computer Applications in Medical Care 1994
 New York: Mc Graw-Hill, Inc 1994 S399
- [Safran 1996 a]
 Safran C., Rind D.M., Daniel Z. et al.
 Development of a Knowledge-Bases Electronic Patient Record
 M.D.Computing Vol 13 No 1 (1996) S46
- [Safran 1996 b]
 Safran C., Rind D.M., et al.
 Effects of a knowledge-based electronic patient record on adherence to practice
 guidelines
 M.D.Computing Vol 13 No 1 (1996) S55
- [Shortliffe 1993]
 Shortliffe E.H.
 The adolescence of AI in Medicine: Will the field become of age in the '90s ?
 Artificial Intelligence in Medicine 5 (1993) S93
- [Shortliffe 1976]
 Shortliffe E.H.
 Computer-Based Medical Consultations: MYCIN
 American Elsevier Publishing Co.Inc. 1976
- [Shwe 1993]
 Michael Shwe, Walter Sujansky, Blackford Middleton
 Reuse of Knowledge Represented in the Arden Syntax
 Proceedings of the 17th AMIA Symposium 1993
- [Skeet 1991]
 Skeet R.G.
 Quality and quality control
 Cancer Registration - Principles and Methods
 IARC Scientific Publications N° 95
 Lyon 1991

- [Spitzer 1994]
Spitzer K., Bürsner S.
Wissensbasierte Systeme in der Medizin
Informationstechnik und Technische Informatik 36 (1994)
R. Oldenbourg Verlag
- [Stürner 1993]
Stürner G.
Oracle7 - Die verteilte semantische Datenbank
dbms publishing 1993
- [Szolovits 1982]
Szolovits P.
Artificial Intelligence and Medicine
Szolovits P. (ed)
Artificial Intelligence in Medicine
Westview Press Inc. 1982
- [UICC 1997]
UICC (Sobin L.H., Wittekind C. eds.)
TNM Classification of Malignant Tumours 5th edition
Wiley & Sons New York 1997
- [Van der Lei 1990]
Van der Lei J., Musen MA
A separation of critiquing knowledge from medical knowledge: Implications for the
Arden Syntax
Proceedings of the IMIA Working Conference on Software Engineering in Medical
Informatics; Timmers T., Blum BI. (ed.)
Amsterdam: North-Holland 1991 S499
- [Van Holten 1991]
Van Holten V.
Editing for consistency of data items
Cancer Registration - Principles and Methods
IARC Scientific Publications N° 95
Lyon 1991
- [Wagner 1983]
Wagner, G., Grundmann, E.
Basisdokumentation für Tumorkranke, 3. Auflage
Springer Verlag Berlin Heidelberg New York 1983
- [WHO 1976]
International Classification of Diseases for Oncology (ICD-O), First Edition
World Health Organization Geneva 1976

[WHO 1990]

Percy C., Van Holten V., Muir C. (eds.)
International Classification of Diseases for Oncology (ICD-O), Second Edition
World Health Organization Geneva 1990

[Wigertz 1986]

Wigertz O.
Making Decisions Based on „Fuzzy“ Medical Data - Can Expert Systems Help ?
Meth. Inform. Med. 25 (1986) S59

[Yu 1979 a]

Yu V.L., Buchanan B.B. et al.
Evaluating the performance of a computer-based consultant
Computer Programs in Biomedicine 9 (1979) S95

[Yu 1979 b]

Yu V.L., Fagan L.M. et al.
Antimicrobial selection by a computer - A blinded evaluation by infectious disease
experts
Journal American Medical Association 242 (1979) S1279

Anhang C: Realisierte Medical Logic Modules

iarc0

/******

maintenance:

title: Incidence/birthdate check;;
filename: iarc0;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
Sicherung der Datenqualität;;
explanation:
Das Diagnosedatum muss einigen grundsätzlichen zeitlichen Kriterien genügen.;;
keywords:
?;
;;
citations:
IARC Technical Report No. 19;
;;

knowledge:

type: DATA_DRIVEN;;

data:

INSERT_TUMOR := EVENT {insert,tumor};
UPDATE_DIA_DATUM := EVENT {update,tumor,diagnosedatum};
DIA_DATUM := READ last{SQL: select diagnosedatum from tumor where fk_patientpat_id=iarc0.fk_patientpat_id and tumor_id = iarc0.tumor_id};
PAT_NAME := READ last{SQL: select name from patient where pat_id = iarc0.fk_patientpat_id};
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = iarc0.fk_patientpat_id};
geburtsdatum := READ last{SQL: select geburtsdatum from patient where pat_id = iarc0.fk_patientpat_id};
;;

;;

priority: 50;;

evoke:

INSERT_TUMOR OR UPDATE_DIA_DATUM;

;;

;;

logic:

dia_datum_fehlt := false;
gr_tagesdatum := false;
gebdat_fehlt := false;
kl_gebdat := false;

if (pat_geschl = "M") then
anrede := "Herrn";
else
anrede := "Frau";
endif;

if (dia_datum is null) then
meldungs_modus := "INFORMATION";
dia_datum_fehlt := true;
conclude true;
endif;

if (dia_datum > tagesdatum) then
meldungs_modus := "FEHLER";
gr_tagesdatum := true;

```

    conclude true;
endif;

if (geburtsdatum is null) then
    meldungs_modus := "INFORMATION";
    gebdat_fehlt := true;
    conclude true;
endif;

if (dia_datum < geburtsdatum) then
    meldungs_modus := "FEHLER";
    kl_gebdat := true;
    conclude true;
endif;

conclude false;
;;

;;

    action:
if dia_datum_fehlt then
    write "Bei "||anrede||" "||pat_name||" wurde noch nicht das Datum der Diagnose dokumentiert !";
endif;

if gr_tagesdatum then
    WRITE "Bei "||anrede||" "||pat_name||" liegt das Diagnosedatum zeitlich NACH dem aktuellen Tagesdatum !";
endif;

if gebdat_fehlt then
    write "Bei "||anrede||" "||pat_name||" wurde noch nicht das Geburtsdatum dokumentiert !";
endif;

if kl_gebdat then
    WRITE "Bei "||anrede||" "||pat_name||" liegt das Diagnosedatum zeitlich VOR dem Geburtsdatum !";
endif;
;;

;;

    urgency: 50;;

end:

```

iarc2

/******

maintenance:

title: Prevents wrong or unlikely age/site/histology combinations;;
filename: iarc2;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
Sicherung der Datenqualität;;
explanation:
IARC-Prüfung zur Erkennung falscher/unwahrscheinlicher Alter/Lokalisation/Histologie Kombinationen;;
keywords:
?;
;;
citations:
IARC Technical Report No. 19;
;;

knowledge:

type: DATA_DRIVEN;;

data:

INSERT_HISTOLOGIEiarc2:= EVENT {insert,histologie,fk_histologie_shis};
UPDATE_HISTOLOGIEiarc2:= EVENT {update,histologie,fk_histologie_shis};
INSERT_LOKALISATIONiarc2:= EVENT {insert,lokalisation,FK_LOKALISATIONLOK};
UPDATE_LOKALISATIONiarc2:= EVENT
{update,lokalisation,FK_LOKALISATIONLOK};
geburtsdatum := READ last{SQL: select geburtsdatum from patient where pat_id = iarc2.fk_tumorfk_patient};

HIST_LISTE_D := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = iarc2.fk_tumorfk_patient and
fk_tumortumor_id = iarc2.fk_tumortumor_id };
LOK_LISTE_D := READ {SQL: select fk_lokalisationlok from lokalisation where fk_tumorfk_patient = iarc2.fk_tumorfk_patient and
fk_tumortumor_id = iarc2.fk_tumortumor_id};
PAT_NAME := READ last{SQL: select name from patient where pat_id = iarc2.fk_tumorfk_patient};
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = iarc2.fk_tumorfk_patient};
;;

;;

priority: 50;;

evoke:

INSERT_HISTOLOGIEiarc2 OR UPDATE_HISTOLOGIEiarc2 OR INSERT_LOKALISATIONiarc2 OR
UPDATE_LOKALISATIONiarc2;

;;

;;

logic:

/*Initialisierungen*/

meldung1 := false;
meldung4 := false;
meldung5 := false;
meldung6 := false;
meldung7 := false;

/*nur fuer aktuelle Auflagen(sous-entendu)*/

if ((LOK_LISTE_D is null) or (HIST_LISTE_D is null)) then
conclude false;
endif;

age := alter(geburtsdatum);

if (pat_geschl = "M") then
anrede := "Herrn";
else
anrede := "Frau";

```

endif;

/*spezifitäten anpassen */
LOK_LISTE_D := SUBSTR(LOK_LISTE_D,1,2);
HIST_LISTE_D := SUBSTR(HIST_LISTE_D,1,4);

/*nun die eigentliche Prüfung*/
/*if age is less than 40 and site is C61 and histology is 814*/
if ((age <= 40) and (lok_liste_d = "61") and (substr(hist_liste_d,1,3) = "814"))
then
    meldung1 := true;
    conclude true;
endif;

/*if age is less than 5 and site is C53 or C61*/

/*if age is less than 20 and site is C17 and histology is less than 9590*/
if ((age < 20) and (lok_liste_d = "17") and (hist_liste_d < "9590"))
then
    meldung4 := true;
    conclude true;
endif;

/*if age is less then 20 and site is C33, C34 or C18 and histology*/
/*is not equal to 824*/
if ( (age < 20) and ( (lok_liste_d = "33") or
                    (lok_liste_d = "34") or
                    (lok_liste_d = "18") )
    and (substr(hist_liste_d,1,3) <> "824") )
then
    meldung5 := true;
    conclude true;
endif;

/*if age is more then 5 and site is C69 and histology is 9510 or 9512*/
if ( (age > 5) and ( (hist_liste_d = "9510") or
                    (hist_liste_d = "9512") )
    and (lok_liste_d = "69") )
then
    meldung6 := true;
    conclude true;
endif;

/*if age is less then 1 or more than 45 and site is C58 and histology is 9100*/
if ( ((age < 15) or
      (age > 45)) and (hist_liste_d = "9100") and (lok_liste_d = "58") )
then
    meldung7 := true;
    conclude true;
endif;
;;
;;

    action:
if meldung1 then
WRITE "Maligne Tumoren der Prostata mit Histologie <Adenokarzinom> sind bei Patienten unter 40 Jahren (Integritätsprüfung IARC) sehr
unwahrscheinlich. ( zu "||anrede||" "||pat_name||" Histologie: "||HIST_LISTE_D||" Lokalisation: "||LOK_LISTE_D||" Alter "||age||")";
endif;

if meldung4 then
WRITE "Maligne Tumoren des Dünndarms ohne Histologie <Lymphom> sind bei Patienten unter 20 Jahren (Integritätsprüfung IARC) sehr
unwahrscheinlich. ( zu "||anrede||" "||pat_name||", Histologie: "||HIST_LISTE_D||" Lokalisation : "||LOK_LISTE_D||" Alter "||age||")";
endif;

if meldung5 then
WRITE "Maligne Tumoren an Dickdarms, Trachea, Lunge oder Bronchus und histologisch kein <Karzinoid> sind bei Patienten unter 20 Jahren
(Integritätsprüfung IARC) sehr unwahrscheinlich. ( zu "||anrede||" "||pat_name||", Histologie: "||HIST_LISTE_D||" Lokalisation :
"||LOK_LISTE_D||" Alter "||age||")";
endif;

if meldung6 then

```

```

WRITE "Maligne Tumoren des Auges und Augenanhanges mit Histologie <Retinoblastom (o.n.A. oder undifferenziert>) sind bei Patienten älter
5 Jahren (Integritätsprüfung IARC) sehr unwahrscheinlich. ( zu "||anrede||" "||pat_name||", Histologie: "||HIST_LISTE_D||" Lokalisation :
"||LOK_LISTE_D||" Alter "||age||")";
endif;

if meldung7 then
WRITE "Maligne Tumoren der Plazenta mit Histologie <Corionkarzinom> sind bei Patienten jünger 15 und älter 45 Jahren (Integritätsprüfung
IARC) sehr unwahrscheinlich. ( zu "||anrede||" "||pat_name||", Histologie: "||HIST_LISTE_D||" Lokalisation : "||LOK_LISTE_D||" Alter "||age||")";
endif;
;;
;;
urgency: 50;;

end:

```

iarc3

/******
maintenance:

title: Prevents wrong or unlikely age/site combinations;;
filename: iarc3;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
Sicherung der Datenqualität;;
explanation:
IARC-Prüfung zur Erkennung falscher/unwahrscheinlicher Alter/Lokalisation Kombinationen;;
keywords:
?;
;;
citations:
IARC Technical Report No. 19;
;;

knowledge:

type: DATA_DRIVEN;;

data:

INSERT_LOKALISATIONiarc3:= EVENT {insert,lokalisation,FK_LOKALISATIONLOK};
UPDATE_LOKALISATIONiarc3:= EVENT
{update,lokalisation,FK_LOKALISATIONLOK};
geburtsdatum := READ last{SQL: select geburtsdatum from patient where pat_id = iarc3.fk_tumorfk_patient};

LOK_LISTE_D := READ {SQL: select fk_lokalisationlok from lokalisation where fk_tumorfk_patient = iarc3.fk_tumorfk_patient and
fk_tumortumor_id = iarc3.fk_tumortumor_id};
LOK_LISTE_D_TEXT := READ {SQL: select ltext from lokalisation where fk_tumorfk_patient = iarc3.fk_tumorfk_patient and
fk_tumortumor_id = iarc3.fk_tumortumor_id};
PAT_NAME := READ last{SQL: select name from patient where pat_id = iarc3.fk_tumorfk_patient};
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = iarc3.fk_tumorfk_patient};
;;
;;

priority: 50;;

evoke:

INSERT_LOKALISATIONiarc3 OR UPDATE_LOKALISATIONiarc3;
;;
;;

logic:

/*Initialisierungen*/
meldung2 := false;
meldung3 := false;

/*nur fuer aktuelle Auflagen(sous-entendu)*/
if (LOK_LISTE_D is null) then
conclude false;
endif;

age := alter(geburtsdatum);

if (pat_geschl = "M") then
anrede := "Herrn";
else
anrede := "Frau";
endif;

/*spezifitäten anpassen */
LOK_LISTE_D := SUBSTR(LOK_LISTE_D,1,2);


```

/*nun die eigentliche Prüfung*/

/*if age is less than 5 and site is C53,C61 */
if ( (age < 5) and ((lok_liste_d = "53") or
                    (lok_liste_d = "61"))) )
then
    meldung2 := true;
    conclude true;
endif;

/*if age is less than 20 and site is C15, C19-25, C50, C53-C55*/
if ( (age < 20) and ((lok_liste_d = "15") or
                    (lok_liste_d = "19") or
                    (lok_liste_d = "20") or
                    (lok_liste_d = "21") or
                    (lok_liste_d = "23") or
                    (lok_liste_d = "23") or
                    (lok_liste_d = "25") or
                    (lok_liste_d = "50") or
                    (lok_liste_d = "53") or
                    (lok_liste_d = "54") or
                    (lok_liste_d = "55"))) )
then
    meldung3 := true;
    conclude true;
endif;
;;
;;

    action:
if meldung2 then
WRITE "Maligne Tumoren an der/n Lokalisation/en ("||LOK_LISTE_D_TEXT||") sind bei Patienten unter 5 Jahren (Integritätsprüfung IARC)
sehr unwahrscheinlich. ( zu "||anrede||" "||pat_name||" Lokalisation : "||LOK_LISTE_D||" Alter "||age||")";
endif;

if meldung3 then
WRITE "Maligne Tumoren an der/n Lokalisation/en ("||LOK_LISTE_D_TEXT||") sind bei Patienten unter 20 Jahren (Integritätsprüfung IARC)
sehr unwahrscheinlich. ( zu "||anrede||" "||pat_name||" Lokalisation : "||LOK_LISTE_D||" Alter "||age||")";
endif;
;;
;;

    urgency: 50;;

end:

```

iarc4

/******

maintenance:

title: Prevents wrong or unlikely age/histology combinations;;
filename: iarc4;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
Sicherung der Datenqualität;;
explanation:
IARC-Prüfung zur Erkennung falscher/unwahrscheinlicher Alter/Histologie Kombinationen;;
keywords:
?;
;;
citations:
IARC Technical Report No. 19;
;;

knowledge:

type: DATA_DRIVEN;;

data:

INSERT_HISTOLOGIEiarc4:= EVENT {insert,histologie,fk_histologie_shis};
UPDATE_HISTOLOGIEiarc4:= EVENT {update,histologie,fk_histologie_shis};

geburtsdatum := READ last{SQL: select geburtsdatum from patient where pat_id = iarc4.fk_tumorfk_patient};

HIST_LISTE_D := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = iarc4.fk_tumorfk_patient and
fk_tumortumor_id = iarc4.fk_tumortumor_id };

HIST_LISTE_D_TEXT := READ {SQL: select htext from histologie where fk_tumorfk_patient = iarc4.fk_tumorfk_patient and
fk_tumortumor_id = iarc4.fk_tumortumor_id };

PAT_NAME := READ last{SQL: select name from patient where pat_id = iarc4.fk_tumorfk_patient};
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = iarc4.fk_tumorfk_patient};
;;

;;

priority: 50;;

evoke:

INSERT_HISTOLOGIEiarc4 OR UPDATE_HISTOLOGIEiarc4;

;;

;;

logic:

/*Initialisierungen*/

meldung1 := false;

meldung2 := false;

/*nur fuer aktuelle Auflagen(sous-entendu)*/

if (HIST_LISTE_D is null) then

conclude false;

endif;

age := alter(geburtsdatum);

if (pat_geschl = "M") then

anrede := "Herrn";

else

anrede := "Frau";

endif;

/*spezifitäten anpassen */

```

HIST_LISTE_D := SUBSTR(HIST_LISTE_D,1,4);

/*nun die eigentliche Prüfung*/
/*if age is less or equal 25 and histology is 9730, 9823, 9863, 9890*/
if ( (age < 26) and ( (hist_liste_d = "9730") or
                    (hist_liste_d = "9823") or
                    (hist_liste_d = "9863") or
                    (hist_liste_d = "9890") ) ) )
then
    meldung1 := true;
    conclude true;
endif;

/*if age is greater or equal 15 and histology is*/
/*8910,8960,8961,8962,8970,8981,8991,9072,9470,9687,9750*/
if ( (age > 14) and ( (hist_liste_d = "8910") or
                    (hist_liste_d = "8960") or
                    (hist_liste_d = "8961") or
                    (hist_liste_d = "8962") or
                    (hist_liste_d = "8970") or
                    (hist_liste_d = "8981") or
                    (hist_liste_d = "8991") or
                    (hist_liste_d = "9072") or
                    (hist_liste_d = "9470") or
                    (hist_liste_d = "9687") or
                    (hist_liste_d = "9750") ) ) )
then
    meldung2 := true;
    conclude true;
endif;
;;
;;

    action:
if meldung1 then
WRITE "Maligne Tumoren mit der/n Histologie/n ("||HIST_LISTE_D_TEXT||") sind bei Patienten unter 26 Jahren (Integritätsprüfung IARC)
sehr unwahrscheinlich. ( zu "||anrede||" "||pat_name||" Histologie : "||HIST_LISTE_D||" Alter "||age||")";
endif;

if meldung2 then
WRITE "Maligne Tumoren mit der/n Histologie/n ("||HIST_LISTE_D_TEXT||") sind bei Patienten über 14 Jahren (Integritätsprüfung IARC)
sehr unwahrscheinlich. ( zu "||anrede||" "||pat_name||" Histologie : "||HIST_LISTE_D||" Alter "||age||")";
endif;
;;
;;

    urgency: 50;;

end:

```

iarc5

/******

maintenance:

title: Impossible sex/site combinations;;
filename: iarc5;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: 1997-12-11;;
validation: TESTING;;

library:

purpose:
Sicherung der Datenqualität;;
explanation:
IARC-Prüfung zur Erkennung falscher Geschlecht/Lokalisation Kombinationen;;
keywords:
?;
;;
citations:
IARC Technical Report No. 19;
;;

knowledge:

type: DATA_DRIVEN;;

data:

```
insert_lokalisation := event{insert,lokalisation,fk_lokalisationlok};
update_lokalisation := event{update,lokalisation,fk_lokalisationlok};
lok_code           := READ last {SQL: select fk_lokalisationlok from lokalisation where fk_tumorfk_patient = iarc5.fk_tumorfk_patient and
fk_tumortumor_id = iarc5.fk_tumortumor_id and fk_lokalisationlok = iarc5.fk_lokalisationlok };
lok_auflage        := READ last{SQL: select fk_lokalisationauf from lokalisation where fk_tumorfk_patient = iarc5.fk_tumorfk_patient and
fk_tumortumor_id = iarc5.fk_tumortumor_id and fk_lokalisationlok = iarc5.fk_lokalisationlok };
geschlecht         := READ last{SQL: select geschlecht from patient where pat_id = iarc5.fk_tumorfk_patient};
pat_name           := READ first 5 from{SQL: select name from patient where pat_id = iarc5.fk_tumorfk_patient};
;;
```

;;

priority: 50;;

evoke:

insert_lokalisation or update_lokalisation;

;;

;;

logic:

lokistwgeschlistq := false;
lokistmgeschlistq := false;

```
if (lok_code is null) or (lok_auflage is null) or (lok_auflage <> "4")
then
  conclude false;
endif;
```

```
if (geschlecht = "M") then
  anrede := "Herrn";
else
  anrede := "Frau";
endif;
```

lok_code = substr(lok_code,1,2);

/*weibliche Lokalisation für männlichen Patienten*/

```
if ( (lok_code = "51") or (lok_code = "52") or
    (lok_code = "53") or (lok_code = "54") or
    (lok_code = "55") or (lok_code = "56") or
    (lok_code = "57") or (lok_code = "58") ) and (geschlecht = "M")
then
```

```
  lokistwgeschlistq := true;
  meldungs_modus := "FEHLER";
  conclude true;
```

```

endif;

/*männliche Lokalisation bei weiblichem Patienten*/
if ( (lok_code = "60") or (lok_code = "61") or
    (lok_code = "62") or (lok_code = "63") ) and (geschlecht = "W")
then
    lokistmgeschlistq := true;
    meldungs_modus := "FEHLER";
    conclude true;
endif;
;;

    action:
if lokistwgeschlistq then
    write "Sie haben bei Herrn "||pat_name||" eine typisch weibliche Lokalisation ("||lok_code||") dokumentiert. Bitte kontrollieren ! (IARC-Integritätsprüfung)";
endif;

if lokistmgeschlistq then
    write "Sie haben bei Frau "||pat_name||" eine typisch männliche Lokalisation ("||lok_code||") dokumentiert. Bitte kontrollieren ! (IARC-Integritätsprüfung)";
endif;
;;

    urgency: 50;;

end:

```

iarc6

/******

maintenance:

title: Prevents wrong or unlikely sex/histology combinations;;
filename: iarc6;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
Sicherung von Datenqualität;;
explanation:
IARC-Prüfung zur Erkennung falscher/unwahrscheinlicher Geschlecht/Histologie Kombinationen;;
keywords:
?;
;;
citations:
IARC Technical Report No. 19;
;;

knowledge:

type: DATA_DRIVEN;;

data:

INSERT_HISTOLOGIEiarc6:= EVENT {insert,histologie,fk_histologie_shis};
UPDATE_HISTOLOGIEiarc6:= EVENT {update,histologie,fk_histologie_shis};

HIST_LISTE_D := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = iarc6.fk_tumorfk_patient and
fk_tumortumor_id = iarc6.fk_tumortumor_id };

HIST_LISTE_D_TEXT := READ {SQL: select htext from histologie where fk_tumorfk_patient = iarc6.fk_tumorfk_patient and
fk_tumortumor_id = iarc6.fk_tumortumor_id };

PAT_NAME := READ last{SQL: select name from patient where pat_id = iarc6.fk_tumorfk_patient};
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = iarc6.fk_tumorfk_patient};
;;

;;

priority: 50;;

evoke:

INSERT_HISTOLOGIEiarc6 OR UPDATE_HISTOLOGIEiarc6;
;;
;;

logic:

/*Initialisierungen*/
meldung1 := false;
meldung2 := false;

/*nur fuer aktuelle Auflagen(sous-entendu)*/
if ((HIST_LISTE_D is null) or (pat_geschl is null)) then
conclude false;
endif;

if (pat_geschl = "M") then
anrede := "Herrn";
else
anrede := "Frau";
endif;

/*spezifitäten anpassen */
HIST_LISTE_D := SUBSTR(HIST_LISTE_D,1,4);

/*nun die eigentliche Prüfung*/

```

/*if sex is male and histology is ...*/
if ( (pat_geschl = "M") and ( (hist_liste_d = "8380") or
    (hist_liste_d = "8381") or
    (hist_liste_d = "8441") or
    (hist_liste_d = "8460") or
    (hist_liste_d = "8461") or
    (hist_liste_d = "8462") or
    (hist_liste_d = "8470") or
    (hist_liste_d = "8471") or
    (hist_liste_d = "8472") or
    (hist_liste_d = "8473") or
    (hist_liste_d = "8600") or
    (hist_liste_d = "8601") or
    (hist_liste_d = "8602") or
    (hist_liste_d = "8610") or
    (hist_liste_d = "8620") or
    (hist_liste_d = "8621") or
    (hist_liste_d = "8622") or
    (hist_liste_d = "8623") or
    (hist_liste_d = "8632") or
    (hist_liste_d = "8660") or
    (hist_liste_d = "8670") or
    (hist_liste_d = "8931") or
    (hist_liste_d = "9000") or
    (hist_liste_d = "9013") or
    (hist_liste_d = "9014") or
    (hist_liste_d = "9015") or
    (hist_liste_d = "9084") or
    (hist_liste_d = "9090") or
    (hist_liste_d = "9091") ) ) )
then
    meldung1 := true;
    conclude true;
endif;

/*if sex is female and histology is 9061, 9062, 9063, 9102*/
if ( (pat_geschl = "W") and ( (hist_liste_d = "9061") or
    (hist_liste_d = "9062") or
    (hist_liste_d = "9063") or
    (hist_liste_d = "9102") ) ) )
then
    meldung2 := true;
    conclude true;
endif;
;;
;;

    action:
if meldung1 then
WRITE "Maligne Tumoren mit der/n Histologie/n ("||HIST_LISTE_D_TEXT||") sind bei MÄNNLICHEN Patienten (Integritätsprüfung IARC)
sehr unwahrscheinlich. ( zu "||anrede||" "||pat_name||" Histologie : "||HIST_LISTE_D||")";
endif;

if meldung2 then
WRITE "Maligne Tumoren mit der/n Histologie/n ("||HIST_LISTE_D_TEXT||") sind bei WEIBLICHEN Patienten (Integritätsprüfung IARC)
sehr unwahrscheinlich. ( zu "||anrede||" "||pat_name||" Histologie : "||HIST_LISTE_D||")";
endif;
;;
;;

    urgency: 50;;

end:

```

iarc10

/******
maintenance:

title: Prevents wrong or unlikely site/histology combinations combinations;;
filename: iarc10;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
?;;
explanation:
IARC-Prüfung zur Erkennung falscher/unwahrscheinlicher Lokalisations/Histologie Kombinationen;;
keywords:
?;
;;
citations:
IARC Technical Report No. 19;
;;

knowledge:

type: DATA_DRIVEN;;

data:

/*Ereignisdefinitionen*/

INSERT_HISTOLOGIEiarc10:= EVENT {insert,histologie,fk_histologie_shis};
UPDATE_HISTOLOGIEiarc10:= EVENT {update,histologie,fk_histologie_shis};

/*gerade dokumentierte Histologie*/

HIST_CODE := READ last{SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = iarc10.fk_tumorfk_patient and
fk_tumortumor_id = iarc10.fk_tumortumor_id and lfdnr = iarc10.lfdnr};

/*Familie, der die doku. Histologie angehoert*/

HIST_FAMILIE := READ last{SQL: select familie from histologie_familie where hist = substr(HIST_CODE,1,4)};

/*alle mit obiger Familie vertraeglichen Lokalisationen*/

LOK_LISTE_W := READ {SQL: select lok from familie_lokalisation_vertr where familie = HIST_FAMILIE };

/*alle dokumentierten Lokalisationen*/

LOK_LISTE_D := READ {SQL: select fk_lokalisationlok from lokalisation where fk_tumorfk_patient = iarc10.fk_tumorfk_patient and
fk_tumortumor_id = iarc10.fk_tumortumor_id};

/*Kosmetik*/

PAT_NAME := READ last{SQL: select name from patient where pat_id = iarc10.fk_tumorfk_patient};
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = iarc10.fk_tumorfk_patient};
HIST_FAMILIE_NAME := READ last{SQL: select beschreibung from familien_beschreibung where schluessel = HIST_FAMILIE};
;;

;;

priority: 50;;

evoke:

INSERT_HISTOLOGIEiarc10 OR UPDATE_HISTOLOGIEiarc10;

;;

;;

logic:

/*nur fuer aktuelle Auflagen(sous-entendu)*/

if ((LOK_LISTE_D is null) or (hist_code is null) or (HIST_FAMILIE is null)) then
conclude false;
endif;

if (pat_geschl = "M") then
anrede := "Herrn";
else
anrede := "Frau";
endif;


```

if (LOK_LISTE_W is null) then
  /*Fall : Histo. ohne Angaben in IARC-Wissensbank*/
  conclude false;
endif;

/*bevor lokalisationslisten verglichen werden, spezifitaet anpassen */
LOK_LISTE_D := SUBSTR(LOK_LISTE_D,1,3);

if (LOK_LISTE_W <> LOK_LISTE_D)
/*es existiert keine Schnittmenge zw. dokumentierten und vertraeglichen L.d.h. geg.histologie mit keiner der Lokalisationen vertraeglich!*/
then
  meldungs_modus := "WARNUNG";
  conclude true;
endif;
;;
;;

  action:
WRITE " Nach den Empfehlungen der IARC ist bei "||anrede||" "||pat_name||" die Histologie (Schlüssel: "||hist_code||" aus Familie:
"||HIST_FAMILIE_NAME||") mit keiner der dok. Lokalisationen (Schlüssel: "||lok_liste_d||") verträglich ! (Prüfung IARC10)";
;;
;;

  urgency: 50;;

end:

```

iarc11

/******

maintenance:

title: Prevents wrong or unlikely site/histology combinations;;
filename: iarc11;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
 unterschied zu iarc10 : basiert auf UNvertraeglichen Kombinationen;;
explanation:
 ?;;
keywords:
 ?;
 ;;
 ;;
citations:
 IARC Technical Report 19;
 ;;

knowledge:

type: DATA_DRIVEN;;

data:

/*Ereignisdefinitionen*/

INSERT_HISTOLOGIEiarc11:= EVENT {insert,histologie,fk_histologie_shis};
UPDATE_HISTOLOGIEiarc11:= EVENT {update,histologie,fk_histologie_shis};

/*gerade dokumentierte Histologie*/

HIST_CODE := READ last{SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = iarc11.fk_tumorfk_patient and
fk_tumortumor_id = iarc11.fk_tumortumor_id and lfdnr = iarc11.lfdnr};

/*Familie, der die doku. Histologie angehört*/

HIST_FAMILIE := READ last{SQL: select familie from histologie_familie where hist = substr(HIST_CODE,1,4)};

/*alle mit obiger Familie UNvertraeglichen Lokalisationen*/

LOK_LISTE_F := READ {SQL: select lok from familie_lokalisation_unvertr where familie = HIST_FAMILIE };

/*alle dokumentierten Lokalisationen*/

LOK_LISTE_D := READ {SQL: select fk_lokalisationlok from lokalisation where fk_tumorfk_patient = iarc11.fk_tumorfk_patient and
fk_tumortumor_id = iarc11.fk_tumortumor_id};

/*Kosmetik*/

PAT_NAME := READ last{SQL: select name from patient where pat_id = iarc11.fk_tumorfk_patient};
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = iarc11.fk_tumorfk_patient};
HIST_FAMILIE_NAME := READ last{SQL: select beschreibung from familien_beschreibung where schluessel = HIST_FAMILIE};
;;
;;

priority: 50;;

evoke:

INSERT_HISTOLOGIEiarc11 OR UPDATE_HISTOLOGIEiarc11;
;;
;;

logic:

/*nur fuer aktuelle Auflagen(sous-entendu)*/

if ((LOK_LISTE_D is null) or (hist_code is null) or (HIST_FAMILIE is null)) then
 conclude false;
endif;

if (pat_geschl = "M") then
 anrede := "Herrn";
else
 anrede := "Frau";
endif;

```

if (LOK_LISTE_F is null) then
  /*Fall : Histo. ohne Angaben in IARC-Wissensbank*/
  conclude false;
endif;

/*bevor lokalisationslisten verglichen werden, spezifitaet anpassen */
LOK_LISTE_D := SUBSTR(LOK_LISTE_D,1,3);

if (LOK_LISTE_F <> LOK_LISTE_D)
  /*es existiert keine Schnittmenge zw. dokumentierten und UNvertraeglichen L.d.h. geg.histologie mit keiner der Lokalisationen UNvertraeglich!*/
  then
    conclude false;
  else
    meldungs_modus := "WARNUNG";
    conclude true;
  endif;
;;

  action:
  WRITE " Nach den Empfehlungen der IARC ist bei "||anrede||" "||pat_name||" die Histologie (Schlüssel: "||hist_code||" aus Familie:
  "||HIST_FAMILIE_NAME||") mit mindestens einer der dok. Lokalisationen (Schlüssel: "||lok_liste_d||") UNverträglich ! (Prüfung IARC11);
  ;;

  urgency: 50;;

end:

```

ip1

/*****

maintenance:

title: kontrolle der Anzahl von Haupthistologien;;
filename: ip1;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: 1995-03-09;;
validation: TESTING;;

library:

purpose:
?;;
explanation:
Gemäß der TBD darf es zu einem Fall nur eine Haupthistologie geben !;;
keywords:
?;
;;
citations:
?;
;;

knowledge:

type: DATA_DRIVEN;;

data:

```
INSERT_HISTOLOGIE_HAUPT := EVENT {insert,histologie,haupt_neben,"H"};  
UPDATE_HISTOLOGIE_HAUPT := EVENT {update,histologie,haupt_neben,"H"};  
ANZ_HAUPT := READ count{SQL: select fk_tumorfk_patient from histologie where fk_tumorfk_patient = ip1.fk_tumorfk_patient and  
fk_tumortumor_id = ip1.fk_tumortumor_id and haupt_neben = 'H'};  
pat_name := READ last{SQL: select name from patient where pat_id = ip1.fk_tumorfk_patient};  
pat_geschl:= READ last{SQL: select geschlecht from patient where pat_id = ip1.fk_tumorfk_patient};  
diagnose_text := READ last{SQL: select diagnosetext from tumor where fk_patientpat_id = ip1.fk_tumorfk_patient and tumor_id =  
ip1.fk_tumortumor_id};  
;;  
;;
```

priority: 1;;

evoke:

INSERT_HISTOLOGIE_HAUPT or UPDATE_HISTOLOGIE_HAUPT

;

;;

;;

logic:

```
if (pat_geschl = "M") then  
    anrede := "Herrn";  
else  
    anrede := "Frau";  
endif;
```

```
IF ANZ_HAUPT > 1 THEN  
meldungs_modus := "FEHLER";  
CONCLUDE TRUE;  
ENDIF;  
;;  
;;
```

action:

```
WRITE "Zum Tumor (Bez.: "||diagnose_text||" Pat.: "||anrede||" "||pat_name||") wurden "||anz_haupt||" Haupthistologien dokumentiert. Im  
Normalfall sollte nur EINE Haupthistologie vorliegen. (Integritätsprüfung 1)";  
;;  
;;
```

urgency: 50;;

end:

ip2

/*****

maintenance:

title: check zeitliche Beziehung Diagnose/Operationsdatum;;
filename: ip2;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: AGT;;
specialist: ;;
date: 1995-03-09;;
validation: TESTING;;

library:

purpose:
?;;
explanation:
In seltenen Ausnahmefällen kann ein Operationsdatum vor dem Diagnosedatum liegen !;;
keywords:
?;
;;
citations:
?;
;;

knowledge:

type: DATA_DRIVEN;;

data:

INSERT_OP_DATUM := EVENT {insert,operation,op_datum};
UPDATE_OP_DATUM := EVENT {update,operation,op_datum};
OP_DATUM := READ last{SQL: select op_datum from operation where fk_tumorfk_patient = ip2.fk_tumorfk_patient and fk_tumortumor_id = ip2.fk_tumortumor_id and op_nummer = ip2.op_nummer};
DIAGNOSEDATUM := READ last{SQL: select diagnosedatum from tumor where fk_patientpat_id = ip2.fk_tumorfk_patient and tumor_id = ip2.fk_tumortumor_id};
PAT_NAME := READ last{SQL: select name from patient where pat_id = ip2.fk_tumorfk_patient};
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = ip2.fk_tumorfk_patient};
;;

;;

priority: 1;;

evoke:

INSERT_OP_DATUM OR UPDATE_OP_DATUM;
;;

;;

logic:

diagnosedatum_fehlt := false;
op_vor_diagnose := false;
diagnostische_op := false;

if (pat_geschl = "M") then
anrede := "Herrn";
else
anrede := "Frau";
endif;

if (diagnosedatum is null) then
diagnosedatum_fehlt := true;
meldungs_modus := "WARNUNG";
conclude true;
else
if (diagnosedatum > op_datum) then
op_vor_diagnose := true;
meldungs_modus := "FEHLER";
conclude true;
else
if (diagnosedatum = op_datum) then
diagnostische_op := true;

```

        meldungs_modus := "INFORMATION";
        conclude true;
    else
        conclude false;
    endif;
endif;
endif;
;;

        ;;

        action:
if diagnosedatum_fehlt then
    write "Bei "||anrede||" "||pat_name||" wurde noch nicht das Datum der Diagnose dokumentiert ! (Integritätsprüfung 1)";
endif;
if op_vor_diagnose then
    WRITE "Bei "||anrede||" "||pat_name||" wurde eine Operation VOR der Diagnosestellung durchgeführt. Bitte kontrollieren ! (Integritätsprüfung 1)";
endif;
if diagnostische_op then
    WRITE "Bei "||anrede||" "||pat_name||" wurde eine Operation mit u.a. diagnostischer Zielsetzung durchgeführt. (Integritätsprüfung 1)";
endif;
;;

        ;;

        urgency: 50;;

end:

```

ip3

/******

maintenance:

title: Vollständigkeit von Daten zu Metastasen;;
filename: ip3;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: ?;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
?;;
explanation:
Metastasedaten können in unterschiedlichen GTDS-Masken in unterschiedlichem Detailgrad beschrieben werden !;;
keywords:
?;
;;
citations:
?;
;;

knowledge:

type: DATA_DRIVEN;;

data:

```
INSERT_TNM := EVENT {insert,tnm,met,"1"};  
UPDATE_TNM := EVENT {update,tnm,met,"1"};  
INSERT_METASTASE := EVENT {insert,metastase};  
UPDATE_METASTASE := EVENT {update,metastase};  
ANZ_TNM_MET := READ count{SQL: select met from tnm where fk_tumorfk_patient = ip3.fk_tumorfk_patient and fk_tumortumor_id = ip3.fk_tumortumor_id and met = '1'};  
ANZ_MET_A := READ count{SQL: select fk_tumorfk_patient from metastase where fk_tumorfk_patient = ip3.fk_tumorfk_patient and fk_tumortumor_id = ip3.fk_tumortumor_id };  
ANZ_MET_B := READ count{SQL : select fk_tumorfk_patient from metastase where fk_tumorfk_patient = ip3.fk_tumorfk_patient and (fk_tumortumor_id = ip3.fk_tumortumor_id or fk_tumortumor_id IS NULL)};  
PAT_NAME := READ last{SQL: select name from patient where pat_id = ip3.fk_tumorfk_patient};  
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = ip3.fk_tumorfk_patient};  
;;  
;;
```

priority: 50;;

evoke:

1 Minute after INSERT_TNM or 1 Minute after UPDATE_TNM or INSERT_METASTASE or UPDATE_METASTASE;

;;

;;

logic:

meldunga := false;

meldungb := false;

```
if (pat_geschl = "M") then  
  anrede := "Herrn";  
else  
  anrede := "Frau";  
endif;
```

/*P.S. ANZ_MET_A ist Anz. der Einträge in Metastase zu Tumor*/

/*P.S. ANZ_MET_B ist Anz. der Einträge in Metastase zu Tumor und ohne*/

/* Tumorzuordnung*/

/*Fall, dass in TNM, aber nicht in Metastase Hinweis auf Metastase*/

if (ANZ_TNM_MET > 0) and (ANZ_MET_B = 0)

then meldunga := true;

meldungs_modus := "WARNUNG";

conclude true;

endif;

/*Fall, dass in Metastase, aber nicht in TNM Hinweis auf Metastase*/

```

if (ANZ_MET_A > 0) and (ANZ_TNM_MET = 0)
then meldungb := true;
    meldungs_modus := "WARNUNG";
    conclude true;
endif;
;;

    action:

if meldunga then
WRITE "Bei "||anrede||" "||pat_name||" wurde zum Tumor ein TNM-Stadium dokumentiert, das auf eine Metastase hinweist. Vergessen Sie bitte
nicht eine detailliertere Metastasenbeschreibung. (Integritätsprüfung 3).";
endif;

if meldungb then
WRITE "Bei "||anrede||" "||pat_name||" wurde zum Tumor eine Metastasenbeschreibung durchgeführt, ohne daß ein Metastasenhinweis in den
TNM-Angaben vorliegt. (Integritätsprüfung 3).";
endif;
;;

    urgency: 50;;

end:

```


ip4

/*

*/

maintenance:

title: Verlaufshistologie muß mit anderen uebereinstimmen;;
filename: ip4;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: 1996-05-24;;
validation: TESTING;;

library:

purpose:
?;;
explanation:
Im Normalfall unterscheiden sich Verlaufshistologien von schon vorhandenen lediglich durch ihre Dignität !;;
keywords:
?;
;;
citations:
?;
;;

knowledge:

type: DATA_DRIVEN;;

data:
INSERT_HISTOLOGIEv4:= EVENT {insert,histologie,fk_histologie_shis};
UPDATE_HISTOLOGIEv4:= EVENT {update,histologie,fk_histologie_shis};
HIST_CODE := READ last{SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = ip4.fk_tumorfk_patient and
fk_tumortumor_id = ip4.fk_tumortumor_id and lfdnr = ip4.lfdnr};
HERKUNFT := READ last {SQL : select herkunft from histologie where fk_tumorfk_patient = ip4.fk_tumorfk_patient and fk_tumortumor_id =
ip4.fk_tumortumor_id and lfdnr = ip4.lfdnr};
HIST_LISTE := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = ip4.fk_tumorfk_patient and
fk_tumortumor_id = ip4.fk_tumortumor_id and lfdnr < ip4.lfdnr};
PAT_NAME := READ last{SQL: select name from patient where pat_id = ip4.fk_tumorfk_patient};
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = ip4.fk_tumorfk_patient};
;;

;;

priority: 50;;

evoke:
INSERT_HISTOLOGIEv4 OR UPDATE_HISTOLOGIEv4;
;;

;;

logic:
if ((hist_code is null) or (HERKUNFT <> "V"))
then conclude false; endif;

if (pat_geschl = "M") then
anrede := "Herrn";
else
anrede := "Frau";
endif;

if (not (substr(hist_code,1,4) in substr(hist_liste,1,4))) then
meldungs_modus := "WARNUNG";
conclude true;
else
conclude false;
endif;
;;

;;

action:
WRITE "Bei "||anrede||" "||pat_name||" ist die Verlaufshistologie (Schlüssel: "||hist_code||") untypischerweise nicht in den schon dokumentierten
Histologien (Schlüssel: "||hist_liste||") enthalten (Integritätsprüfung 4) !";

::

::

urgency: 50;;

end:

ip5

/*-----*/

maintenance:

title: Wurde T1c nur bei Ovarial/Mamakarzinom verwendet ?;;
filename: ip5;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
?;;
explanation:
Verstoß gegen TNM-Klassifikationskonvention;;
keywords:
?;
;;
citations:
?;
;;

knowledge:

type: DATA_DRIVEN;;

data:

```
INSERT_TNM_T1c := EVENT {insert,tnm,t,"1c"};  
UPDATE_TNM_T1c := EVENT {update,tnm,t,"1c"};  
T_CODE := READ last{SQL: select t from tnm where fk_tumorfk_patient = ip5.fk_tumorfk_patient and fk_tumortumor_id =  
ip5.fk_tumortumor_id and lfdnr = ip5.lfdnr};  
LOK_CODE := READ {SQL: select fk_lokalisierunglok from lokalisierung where fk_tumorfk_patient = ip5.fk_tumorfk_patient and  
fk_tumortumor_id = ip5.fk_tumortumor_id};  
LOK_AUF := READ last {SQL: select fk_lokalisierungauf from lokalisierung where fk_tumorfk_patient = ip5.fk_tumorfk_patient and  
fk_tumortumor_id = ip5.fk_tumortumor_id};  
LOK_TEXT := READ {SQL: select ltext from lokalisierung where fk_tumorfk_patient = ip5.fk_tumorfk_patient and fk_tumortumor_id =  
ip5.fk_tumortumor_id};  
PAT_NAME := READ last{SQL: select name from patient where pat_id = ip5.fk_tumorfk_patient};  
PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = ip5.fk_tumorfk_patient};  
;;  
;;
```

priority: 50;;

evoke:

```
INSERT_TNM_T1c or UPDATE_TNM_T1c;  
;;  
;;
```

logic:

```
/*Kontrolle auf Vorhandensein von TNM_T noetig wegen Aufruf aus*/  
/*Anwendung Zentrale. Hier eventuell aufgerufen, nach Korrektur von T*/  
if ( (lok_code is NULL) or (lok_auf is NULL) or (T_CODE <> "1c") )  
then conclude false; endif;
```

```
if (pat_geschl = "M") then  
  anrede := "Herrn";  
else  
  anrede := "Frau";  
endif;
```

```
if (lok_auf = "3") then  
  if (not( ("1830" in (substr(lok_code,1,4))) or  
    ("174" in (substr(lok_code,1,3))) or  
    ("175" in (substr(lok_code,1,3)))  
  ))  
  then
```

```
    meldungs_modus := "FEHLER";  
    conclude true;
```

```

else
    conclude false;
endif;
endif;

if (lok_auf = "4") then
    if (not( ("56" in (substr(lok_code,1,2))) or
            ("50" in (substr(lok_code,1,2))) ) )
    then
        meldungs_modus := "FEHLER";
        conclude true;
    else
        conclude false;
    endif;
endif;
;;

        ;;

        action:
WRITE "Bei "||anrede||" "||pat_name||" liegt ein Tumor an folgenden Lokalisationen vor : "||lok_text||". Der verwendete TNM-Code T1c ist
eigentlich nur bei Ovarial/ und Mamakarzinomen vorgesehen. (Integritätsprüfung 5)";
;;

        ;;

        urgency: 50;;

end:

```

ip6

/*

*/

maintenance:

title: Verwendung von ANN-ARBOR nur bei Lymphomen;;
filename: ip6;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: A.G.Tafazzoli;;
specialist: ;;
date: 1996-11-14;;
validation: TESTING;;

library:

purpose:

?;;

explanation:

Die ANN-ARBOR Stadienklassifikation sollte nur bei Systemerkrankungen (Lymphome, Leukämien) verwendet werden

!;;

keywords:

?;

;;

citations:

?;

;;

knowledge:

type: DATA_DRIVEN;;

data:

insert_ann_arbor := event{insert,ann_arbor};

update_ann_arbor := event{update,ann_arbor};

hist_liste := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = ip6.fk_tumorfk_patient and
fk_tumortumor_id = ip6.fk_tumortumor_id};

pat_name := READ last{SQL: select name from patient where pat_id = ip6.fk_tumorfk_patient};

ann_arbor_toto := READ last{SQL: select fk_tumorfk_patient from ann_arbor where fk_tumorfk_patient = ip6.fk_tumorfk_patient and
fk_tumortumor_id = ip6.fk_tumortumor_id and lfdnr = ip6.lfdnr};

;;

;;

priority: 50;;

evoke:

insert_ann_arbor or update_ann_arbor;

;;

;;

logic:

meldungz := false;

meldunga := false;

meldungb := false;

if (ann_arbor_toto is null)

then

meldungz:=true;

conclude true;

endif;

if (hist_liste is null)

then

meldunga := true;

conclude true;

endif;

if (not((substr(hist_liste,1,3))>"958")) then

meldungb := true;

conclude true;

endif;

```

conclude false;
;;

        action:
if meldungz then
    write "Ann-Arbor-Eintrag wurde zwischenzeitlich gelöscht !";
endif;

if meldunga then
    write "Beim Patienten "||pat_name||" haben Sie eine ANN-ARBOR Stadienklassifikation durchgeführt, ohne daß Histologien vorliegen. Nach Möglichkeit nachholen ! (Integritätsprüfung 6)";
endif;

if meldungb then
    write "Sie haben beim Patienten "||pat_name||" eine ANN-ARBOR Stadienklassifikation durchgeführt, ohne daß eine lymphomspezifische Histologie ("||hist_list||") vorliegt. Bitte kontrollieren ! (Integritätsprüfung 6)";
endif;
;;

        ;;

        urgency: 50;;

end:

```

ip7

/******

maintenance:

title: Ausschluss von Mehrfachdokumentation EINES Tumors;;
filename: ip7;;
version: ?;;
institution: ?;;
author: A.G. Tafazzoli;;
specialist: ;;
date: august 97;;
validation: TESTING;;

library:

purpose:
?;;
explanation:
Nach der IARC-Richtlinie sollten unterschiedl. Tumoren sich in Histologie oder Lokalisation unterscheiden;;
keywords:
?;
;;
citations:
Cancer Registration Principles and Methods (IARC No 95);
;;

knowledge:

type: ?;;

data:

INSERT_TUMOR_ID:= EVENT {insert,tumor,tumor_id};
UPDATE_TUMOR_ID:= EVENT {update,tumor,tumor_id};

ANZ_TUMOR := READ count{SQL: select tumor_id from tumor where fk_patientpat_id = ip7.fk_patientpat_id};

TUMOR_NR := READ last{SQL: select tumor_id from tumor where fk_patientpat_id = ip7.fk_patientpat_id and tumor_id = ip7.tumor_id};

LOK_LISTE_AKT := READ {SQL: select fk_lokalisationlok from lokalisation where fk_tumorfk_patient = ip7.fk_patientpat_id and fk_tumortumor_id = ip7.tumor_id};

LOK_LISTE_VOR := READ {SQL: select fk_lokalisationlok from lokalisation where fk_tumorfk_patient = ip7.fk_patientpat_id and fk_tumortumor_id <> ip7.tumor_id};

HIST_LISTE_AKT := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = ip7.fk_patientpat_id and fk_tumortumor_id = ip7.tumor_id};

HIST_LISTE_VOR := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = ip7.fk_patientpat_id and fk_tumortumor_id <> ip7.tumor_id};

PAT_NAME := READ last{SQL: select name from patient where pat_id = ip7.fk_patientpat_id};

PAT_GESCHL := READ last{SQL: select geschlecht from patient where pat_id = ip7.fk_patientpat_id};

;;

;;

priority: 50;;

evoke:

1 Minute after INSERT_TUMOR_ID or 1 Minute after UPDATE_TUMOR_ID;

;;

;;

logic:

/*Initialisierungen*/

flag_unerfuellt := false;

flag_erfuellt := false;

flag_womoeglich := false;

flag_tumor_geloescht := false; /*wenn ueber zentrale.fmx aufgerufen*/

text2 := "(Erläuterung : ";

zaehler_unerfuellt := 0; /*keine Meldung*/

zaehler_erfuellt := 0; /*Meldung*/

zaehler_womoeglich := 0; /*Zaehler fuer unentschiedene Bedingungen*/

/*Grundlegende Kontrolle*/

if (anz_tumor = 1) then /*anz_tumor zählt alle Tumoren*/

conclude false;

```

endif;
if (TUMOR_NR is null) then
    flag_tumor_geloescht := true;
    meldungs_modus := "INFORMATION";
    conclude true;
endif;

if (pat_geschl = "M") then
    anrede := "Herr";
else
    anrede := "Frau";
endif;

/*1.Bedingung : akt. Lok. in bisherigen enthalten ?*/
LOK_LISTE_AKT := substr(LOK_LISTE_AKT,1,3);
LOK_LISTE_VOR := substr(LOK_LISTE_VOR,1,3);
if (LOK_LISTE_AKT is null) or (LOK_LISTE_VOR is null) then
    zaehler_womoeglich := zaehler_womoeglich + 1;
    text2:=text2||"keine Lokalisationen dokumentiert ";
else
    if (LOK_LISTE_AKT <> LOK_LISTE_VOR) then
        /*verschieden*/
        zaehler_unerfuellt:= zaehler_unerfuellt + 1;
        text2:=text2||"unterschiedl.Lokalisationen ";
    else
        zaehler_erfuellt:= zaehler_erfuellt + 1;
        text2:=text2||"identische Lokalisationen ";
    endif;
endif;

/*2.Bedingung : akt. Hist. in bisherigen enthalten ?*/
HIST_LISTE_AKT := substr(HIST_LISTE_AKT,1,4);
HIST_LISTE_VOR := substr(HIST_LISTE_VOR,1,4);
if (HIST_LISTE_AKT is null) or (HIST_LISTE_VOR is null) then
    zaehler_womoeglich := zaehler_womoeglich + 1;
    text2:=text2||",keine Histologien dokumentiert )";
else
    if (HIST_LISTE_AKT <> HIST_LISTE_VOR) then
        /*verschieden*/
        zaehler_unerfuellt:= zaehler_unerfuellt + 1;
        text2:=text2||",unterschiedl.Histologien )";
    else
        zaehler_erfuellt:= zaehler_erfuellt + 1;
        text2:=text2||",identische Histologien )";
    endif;
endif;

/*Endeverarbeitung*/
if (zaehler_unerfuellt > 0) then
    flag_unerfuellt := true;
    meldungs_modus := "INFORMATION";
    conclude true;
endif;
if (zaehler_erfuellt = 2) then
    flag_erfuellt := true;
    meldungs_modus := "FEHLER";
    conclude true;
endif;
if (zaehler_womoeglich + zaehler_erfuellt = 2) then
    flag_womoeglich := true;
    meldungs_modus := "WARNUNG";
    conclude true;
endif;

;;

;;

action:
if flag_tumor_geloescht then
    write "Der spezifizierte Tumor wurde zwischenzeitlich geloescht !";
endif;

```



```

if flag_erfuellt then
    write "Gemäß der Richtlinie zur Erkennung von Tumormehrfachdokumentation der IARC ist bei Patient "||pat_name||" der Tumor
    "||TUMOR_NR || " DOPPELT dokumentiert ! "||text2;
endif;

if flag_unerfuellt then
    write "Bei Patient "||pat_name||" zu Tumor "||TUMOR_NR || " gemäß IARC KEIN Problem mit Tumormehrfachdokumentation "||text2;
endif;

if flag_womoeglich then
    write "Bei Patient "||pat_name||" zu Tumor "||TUMOR_NR || " kann eine Tumormehrfachdokumentation gemäß IARC NICHT ausgeschlossen
    werden ! "||text2;
endif;
;;

;;

urgency: 50;;

end:

```

ST_IKNHL

/******

maintenance:

title: Studienpruefung NHL-Studie;;
filename: ST_IKNHL;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: ?;;
specialist: ;;
date: ?;;
validation: TESTING;;

library:

purpose:
?;;
explanation:
?;;
keywords:
;;
citations:
?;
;;

knowledge:

type: DATA_DRIVEN;;

data:

```
INSERT_HISTOLOGIE_NEVER2 := EVENT { update,histologie,haupt_neben,"N"};
HISTOLOGIE_LISTE := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = ST_IKNHL.fk_tumorfk_patient
and fk_tumortumor_id = ST_IKNHL.fk_tumortumor_id};
geburtsdatum := READ last{SQL: select geburtsdatum from patient where pat_id = ST_IKNHL.fk_tumorfk_patient};
pat_name := READ last{SQL: select name from patient where pat_id = ST_IKNHL.fk_tumorfk_patient};
pat_geschl:= READ last{SQL: select geschlecht from patient where pat_id = ST_IKNHL.fk_tumorfk_patient};
;;
```

;;

priority: 50;;

evoke:

INSERT_HISTOLOGIE_NEVER2;

;;

logic:

```
if (pat_geschl = "M") then
  anrede := "Herr";
else
  anrede := "Frau";
endif;
```

/*Variablen und Zaehler*/

```
flag_unerfuellt := false;
flag_erfuellt := false;
flag_womoeglich := false;
text2 := "Beschreibung der Bedingungen";
zaehler_unerfuellt := 0; /*Zaehler fuer unerfuellte Bedingungen*/
zaehler_erfuellt := 0; /*Zaehler fuer erfuellte Bedingungen*/
zaehler_womoeglich := 0; /*Zaehler fuer unentschiedene Bedingungen*/
```

/*1. Bedingung : Alter zw. 18 und 60*/

```
age:=alter(geburtsdatum);
if ( (age < 18) and (age <> 0) ) then
  text2 := "juenger als 18(Ausschluss)";
  zaehler_unerfuellt := zaehler_unerfuellt + 1;
else
  if (age > 60) then
    text2 := "Alter ueber 60(Ausschluss)";
    zaehler_unerfuellt := zaehler_unerfuellt + 1;
  else
    if ( (age < 61) and (age > 17) ) then
```

```

        text2 := "Alter zw. 18 und 60(OK)";
        zaehler_erfuellt := zaehler_erfuellt + 1;
    else
        text2 := "Alter undokumentiert";
        zaehler_womoeglich := zaehler_womoeglich + 1;
    endif;
endif;
endif;

/*2. Bedingung : eine der Histo. 9591 liegt vor*/
histologie_liste:= substr(histologie_liste,1,4); /*nun Dignitaet weg*/
if (histologie_liste = "9591")
then text2:=text2||", NHLymphom (OK)";
    zaehler_erfuellt := zaehler_erfuellt + 1;
else
    text2:=text2||", kein NHLymphom(Ausschluss)";
    zaehler_unerfuellt := zaehler_unerfuellt + 1;
endif;

/*Endeverarbeitung*/
if (zaehler_unerfuellt > 0) then
    flag_unerfuellt := true;
    meldungs_modus := "WARNUNG";
    conclude true;
endif;
if (zaehler_erfuellt = 2) then
    flag_erfuellt := true;
    meldungs_modus := "WARNUNG";
    conclude true;
endif;
if (zaehler_womoeglich + zaehler_erfuellt = 2) then
    flag_womoeglich := true;
    meldungs_modus := "WARNUNG";
    conclude true;
endif;
;;

;;

    action:
    if flag_unerfuellt then
        WRITE "Studie04:" || anrede || " || pat_name ||" erfuellt NICHT die Kriterien ( " || text2 || " )";
    endif;

    if flag_erfuellt then
        WRITE "Studie04:" || anrede || " || pat_name ||" erfuellt die Kriterien ( " || text2 || " )";
    endif;

    ;;

    urgency: 50;;

end:

```

ST_SMSGEM1

/******

maintenance:

title: Studienpruefung SMS-GEM-1;;
filename: ST_SMSGEM1;;
version: 1.00;;
institution: Institut fuer Medizinische Informatik Giessen;;
author: AGT;;
specialist: ;;
date: 1996-11-13;;
validation: TESTING;;

library:

purpose:
?;;
explanation:
?;;
keywords:
;;
citations:
;;

knowledge:

type: DATA_DRIVEN;;

data:

```
INSERT_HISTOLOGIE_NEVER := EVENT {update,histologie,haupt_neben,"N"};
HISTOLOGIE_LISTE := READ {SQL: select fk_histologie_shis from histologie where fk_tumorfk_patient = st_smsgem1.fk_tumorfk_patient
and fk_tumortumor_id = st_smsgem1.fk_tumortumor_id};
geburtsdatum := READ last{SQL: select geburtsdatum from patient where pat_id = st_smsgem1.fk_tumorfk_patient};
anz_chemotherapie := READ count {SQL: select fk_patientpat_id from internistische_therapie where fk_patientpat_id =
st_smsgem1.fk_tumorfk_patient and fk_tumortumor_id = st_smsgem1.fk_tumortumor_id};
anz_metastase_bra := READ count {SQL: select fk_tumorfk_patient from metastase where fk_tumorfk_patient =
st_smsgem1.fk_tumorfk_patient and fk_tumortumor_id = st_smsgem1.fk_tumortumor_id and fk_lokalisationlok = 'BRA'};
pat_name := READ last{SQL: select name from patient where pat_id = st_smsgem1.fk_tumorfk_patient};
pat_geschl:= READ last{SQL: select geschlecht from patient where pat_id = st_smsgem1.fk_tumorfk_patient};
;;
```

;;

priority: 50;;

evoke:

INSERT_HISTOLOGIE_NEVER;

;;

;;

logic:

```
if (pat_geschl = "M") then
  anrede := "Herr";
else
  anrede := "Frau";
endif;
```

/*Variablen und Zaehler*/

flag_unerfuellt := false;

flag_erfuellt := false;

flag_womoeglich := false;

text2 := "Beschreibung der Bedingungen";

zaehler_unerfuellt := 0; /*Zaehler fuer unerfuellte Bedingungen*/

zaehler_erfuellt := 0; /*Zaehler fuer erfuellte Bedingungen*/

zaehler_womoeglich := 0; /*Zaehler fuer unentschiedene Bedingungen*/

/*1. Bedingung : Alter mindestens 18*/

age:=alter(geburtsdatum);

if ((age < 18) and (age <> 0)) then

text2 := "jünger als 18(Ausschluss)";

zaehler_unerfuellt := zaehler_unerfuellt + 1;

else

if (age > 17) then

text2 := "Alter ueber 18(OK)";

zaehler_erfuellt := zaehler_erfuellt + 1;

```

else
    text2 := "Alter undokumentiert";
    zaehler_womoeglich := zaehler_womoeglich + 1;
endif;
endif;

/*2. Bedingung : keine Chemotherapie*/
if (anz_chemotherapie >0)
then
    text2:= text2||", Chemotherapie(Ausschluss)";
    zaehler_unerfuellt := zaehler_unerfuellt + 1;
else text2:= text2||", keine Chemotherapie(OK)";
    zaehler_erfuellt := zaehler_erfuellt + 1;
endif;

/*3. Bedingung : keine Hirnmetasen*/
if (anz_metastase_bra >0)
then
    text2:=text2||", Hirnmetastase(Ausschluss)";
    zaehler_unerfuellt := zaehler_unerfuellt + 1;
else text2:=text2||", keine Hirnmetastasen (OK)";
    zaehler_erfuellt := zaehler_erfuellt + 1;
endif;

/*4. Bedingung : eine der Histo. 8152,8151,8153 liegt vor*/
histologie_liste:= substr(histologie_liste,1,4); /*nun Dignitaet weg*/
if ((histologie_liste = "8152") or
    (histologie_liste = "8151") or
    (histologie_liste = "8153"))
then text2:=text2||", Pankreaskarzinom(OK)";
    zaehler_erfuellt := zaehler_erfuellt + 1;
else
    text2:=text2||", kein Pankreaskarzinom(Ausschluss)";
    zaehler_unerfuellt := zaehler_unerfuellt + 1;
endif;

/*Endeverarbeitung*/
if (zaehler_unerfuellt > 0) then
    flag_unerfuellt := true;
    meldungs_modus := "WARNUNG";
    conclude true;
endif;
if (zaehler_erfuellt = 4) then
    flag_erfuellt := true;
    meldungs_modus := "WARNUNG";
    conclude true;
endif;
if (zaehler_womoeglich + zaehler_erfuellt = 4) then
    flag_womoeglich := true;
    meldungs_modus := "WARNUNG";
    conclude true;
endif;
;;

;;

    action:
if flag_unerfuellt then
WRITE "Studie03:"||anrede||" "||pat_name||" erfuehlt NICHT die Kriterien ( "||text2||")";
endif;

if flag_erfuellt then
WRITE "Studie03:"||anrede||" "||pat_name||" erfuehlt die Kriterien ( "||text2||")";
endif;

if flag_womoeglich then
write "Studie03:"||anrede||" "||pat_name||" noch unvollständig dokumentiert ( "||text2||")";
endif;
;;

;;

    urgency: 50;;
end:

```

Anhang D : Tabellen mit dem IARC-Fachwissen über Lokalisations-Histologieverträglichkeiten

FAMILIEN_BESCHREIBUNG

SQL> desc FAMILIEN_BESCHREIBUNG

Name	Null?	Type

SCHLUESSEL		VARCHAR2(3)
BESCHREIBUNG		VARCHAR2(60)

FAMILIE_LOKALISATION_UNVERTR

SQL> desc FAMILIE_LOKALISATION_UNVERTR

Name	Null?	Type

FAMILIE		VARCHAR2(3)
LOK		VARCHAR2(5)
AUFLAGE		VARCHAR2(3)

FAMILIE_LOKALISATION_VERTR

SQL> desc FAMILIE_LOKALISATION_VERTR

Name	Null?	Type

FAMILIE		VARCHAR2(3)
LOK		VARCHAR2(5)
AUFLAGE		VARCHAR2(3)

HISTOLOGIE_FAMILIE

SQL> desc HISTOLOGIE_FAMILIE

Name	Null?	Type

HIST		VARCHAR2(5)
AUFLAGE		VARCHAR2(3)
FAMILIE		VARCHAR2(3)

HIST_ALL_VERTR

SQL> desc HIST_ALL_VERTR

Name	Null?	Type

HIST		VARCHAR2(5)
AUFLAGE		VARCHAR2(3)
BESCHREIBUNG		VARCHAR2(80)

Anhang E : Schnittstelle des Prozedurpaketes ARDEN_PACK

```
type tp_tabelle is table of varchar2(255)
    index by binary_integer;
type tp_tabelle_date is table of date
    index by binary_integer;

procedure eintrag_tabelle
    (tabelle IN out arden_pack.tp_tabelle, wert in varchar2);
procedure eintrag_tabelle
    (tabelle IN out arden_pack.tp_tabelle_date, wert in date);

procedure ausgabe_tabelle
    (tabelle IN arden_pack.tp_tabelle);
procedure ausgabe_tabelle
    (tabelle IN arden_pack.tp_tabelle_date);
function ausgabe_tabelle_string
    (tabelle IN arden_pack.tp_tabelle)
    return varchar2;
function ausgabe_tabelle_string
    (tabelle IN arden_pack.tp_tabelle_date)
    return varchar2;

procedure debug;

function maximum (tabelle in arden_pack.tp_tabelle)
    return varchar2;
function maximum (tabelle in arden_pack.tp_tabelle_date)
    return date;

function minimum (tabelle in arden_pack.tp_tabelle)
    return varchar2;
function minimum (tabelle in arden_pack.tp_tabelle_date)
    return date;

function member (element in varchar2, tabelle in arden_pack.tp_tabelle)
    return boolean;
function member (element in date, tabelle in arden_pack.tp_tabelle_date)
    return boolean;

function neutral(tabelle in arden_pack.tp_tabelle)
    return arden_pack.tp_tabelle;
function neutral(tabelle in arden_pack.tp_tabelle_date)
    return arden_pack.tp_tabelle_date;

procedure zuweisung(atom in out varchar2, element in varchar2);
procedure zuweisung(tabelle1 in out arden_pack.tp_tabelle,tabelle2 in arden_pack.tp_tabelle);
procedure zuweisung(tabelle in out arden_pack.tp_tabelle, element in varchar2);
procedure zuweisung(tabelle in out arden_pack.tp_tabelle_date, element in date);
procedure zuweisung(atom in out boolean, element in boolean);
procedure zuweisung(atom in out number, element in number);
```

```

procedure zuweisung(atom in out number, element in number);

function ausgabe(atom in varchar2)
    return varchar2;
function ausgabe(tabelle in arden_pack.tp_tabelle)
    return varchar2;
function ausgabe(tabelle in arden_pack.tp_tabelle_date)
    return varchar2;
function ausgabe(atom in number)
    return varchar2;

function is_null(atom in varchar2)
    return boolean;
function is_null(tabelle in arden_pack.tp_tabelle)
    return boolean;
function is_null(tabelle in arden_pack.tp_tabelle_date)
    return boolean;
function is_null(atom in date)
    return boolean;
function is_null(atom in number)
    return boolean;

function vergleich(operator in varchar2, atom1 in varchar2, atom2 in varchar2)
    return boolean;
function vergleich(operator in varchar2, atom1 in number, atom2 in number)
    return boolean;
function vergleich(operator in varchar2, atom1 in number, atom2 in number)
    return boolean;
function vergleich(operator in varchar2, tabelle in arden_pack.tp_tabelle, atom in varchar2)
    return boolean;
function vergleich(operator in varchar2, tabelle1 in arden_pack.tp_tabelle, tabelle2 in
arden_pack.tp_tabelle)
    return boolean;
function vergleich(operator in varchar2, tabelle in arden_pack.tp_tabelle_date,atom in date)
    return boolean;
function vergleich(operator in varchar2, atom1 in date, atom2 in date)
    return boolean;
function vergleich(operator in varchar2, atom1 in boolean, atom2 in boolean)
    return boolean;

function substring(atom in varchar2, number1 in number, number2 in number)
    return varchar2;
function substring(tabelle in arden_pack.tp_tabelle, number1 in number, number2 in number)
    return arden_pack.tp_tabelle;

function age(datum IN date)
    return number;

```


Tabellarischer Lebenslauf

<i>Name</i>	Ali Gerhard TAFAZZOLI-YAZDI
<i>Geburtsdatum, Ort</i>	11. Februar 1968 in Calw / Württemberg
<i>Nationalität</i>	deutsch
<i>Schulausbildung</i>	Grundschule in Tübingen Orientierungsstufe und Gymnasium in Leer / Ostfriesland
<i>Staatsdienst</i>	9 Monate Bundeswehr 11 Monate Zivildienst
<i>Hauptstudium</i>	von 1988 - 1993 : Medizinische Informatik an der Universität Heidelberg Diplomarbeitsthema : „Verbesserter Zugriff auf medizinische Informationsquellen durch Benutzung des UMLS-Metathesaurus und semantischen Netzes sowie konzeptueller Graphen“
<i>Auslandsstudium</i>	von 1992 - 1993 : „Diplome d’Etudes Superieures Specialisees en Intelligence Artificielle et Ingenierie de la Connaissance“ an der Universite de Savoie / Chambéry / Frankreich
<i>Wissenschaftliche Tätigkeit</i>	seit 1994 am Institut für Medizinische Informatik der Justus- Liebig-Universität Tätigkeitsschwerpunkte : Klinische Krebsregister, Datenkommunikation, Systeme zur Entscheidungsunterstützung, XML

Danksagung

Mein herzlicher Dank gilt vor allem Herrn Professor Dr. J. Dudeck für die Überlassung des Themas der Doktorarbeit und für das in mich gesetzte Vertrauen.

Weiterhin möchte ich mich bei meinen langjährigen Arbeitskollegen Herrn Dr. U. Altmann, Herrn F. Katz und Herrn Dr. W. Wächter für die Hilfestellungen in vielen Bereichen bedanken.